

IJUtilities



by Micono Utilities



更新内容

0.9.9d10 (2020/5/3)

新規：MixJuice / MicJackのSoftAPに接続し、UDPでデータ送信が可能になった。



0.9.9d10 (2020/3/3)

修正：Linux版で読みづらくなっている文字を少しみやすく修正した。

修正：ファーム更新 (lpc21isp, ESPTool) の不具合を修正した。

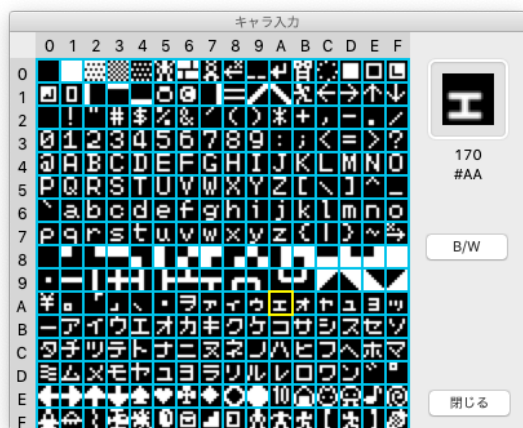
0.9.9d8&9 (2019/12/3)

修正：Win版のMonitor機能でBackspaceキーで2文字消えてしまう不具合を修正。

修正：Monitor機能でinsert状態での挙動の不具合を修正。

改良：Monitor機能でdeleteキー機能に対応。

改良：Monitor機能で文字一覧ウインドウから選択し、入力できるようになった (Monitorウインドウをクリックすると文字一覧ウインドウが表示します)。



重要

●**フォントについて**：同梱されてる「Font」フォルダにIchigoJam用、IchigoLatte用のフォントが入っています。IJUtilitiesを動かすために、それらのフォントをインストールして置く必要があります。

最低以下の4種類のフォントを入れてください。

- IchigoJam-for-Display.ttf
- IchigoJam-for-Display-1.2.ttf
- IchigoLatte-for-Display.ttf
- IchigoQuest-for-Display.ttf

インストール後、IJUtilitiesを起動したら、環境設定のフォントの項目で、使用するフォントを以下の図の様に設定してください。



【目次】 ページの数値をクリックするとそのページにジャンプします

フォントのインストールと設定（重要）	5
配線	6
通信	6
ウインドウの説明	7
《ツールウインドウ》	7
●新規Jamボタン、新規Latteボタン、新規Questボタン	7
●開くボタン	7
●リストビューアボタン	7
●ROMリスト	7
●読み込みボタン	7
●ダウンロードボタン	8
●Firm書換えボタン（旧Ipc21ispボタン）	8
●コマンド一覧ボタン	8
●ピン配列ボタン	9
●リファレンスボタン	9
●環境設定ボタン	9
《ターミナルウインドウ》	12
●シリアル接続	12
●ファンクション・キャラ入力ボタン	12
●App & Link ボタン	12
●IchigoJam Monitorボタン	13
●PanCake シミュレーター	13
●MixJuice 機能ボタン	13
●クリップ送信ボタン	13
●16進入力フィールド・Hex送信	13
●ターミナルフィールド	13
●消去ボタン左のチェックボックス	13
●消去ボタン受送信	13
●UART1,2,3	13
●行間1.0/1.25/1.5	13
《エディタウインドウ》	14
●保存・出力ボタン	14
●読み込みボタン	14
●すべてを送信、選択送信ボタン	14
●IchigonQuestの場合の読み込み（うけとり）と送信（おくる）ボタン	14

●特殊な文字の入力	15
●残りバイト数の表示	15
＊IchigoLatte未対応	15
●折返し表示	15
●白黒反転表示（BW）	15
●シンタックスカラー（C）	15
＊Windows未対応	15
●ファンクションの「RENUM」	15
●変数、配列の一覧と変数の検索置換（IchigoJamのみ、IchigoLatte未対応）	15
●プログラム内容のリスト表示	16
●micro:bit Javascriptとしてコピー	16
【DENUM/RENUM機能】	17
【IchigoJamモニター機能】	18
【IchigoJam Ap版・PanCake, MixJuiceのシミュレータの連携】	20
【ネットを介したファイルの読み書き】	22
【FlashAirファイルツール】	24
【PanCake command Assistant】	25
【マシン語変換（Bin2Pokeの設定及び使い方）】	26
【PCGエディタ】	29
【MixJuiceなどESP-WROOM-02のファームを書き込む】	31
【AQM1248AとIchigoJamをつなぐ】	32
【IchigonQuest用のBASICについて】	33
【micro:bit用のプログラムのコマンドリファレンス】	34
【MixJuice/MicJackのSoftAPに接続しUDPでデータを送信する】	38
免責など	40
主な公開日	40
お問い合わせ	40
過去の更新内容	41

フォントのインストールと設定（重要）

IJUtilitiesはIchigoJamの特殊な文字を表示させる為、特別なフォントをインストールおよび設定する必要があります。

フォントのインストール

v1.1以前をご利用の場合：

同梱された中に「**FONT**」という名前のフォルダがあります。その中の「**for IJ1.1**」のフォルダの中に「IchigoJam-for-Display.ttf」というファイルが入ってます。そのファイルをダブルクリックしてインストール指示に従ってください。他の言語として、モンゴル語のフォントも入ってます。

v1.2以降をご利用の場合：

同梱された中に「**FONT**」という名前のフォルダがあります。その中の「**for IJ1.2**」のフォルダの中に「**IchigoJam-for-Display-1.2.ttf**」というファイルが入ってます。そのファイルをダブルクリックしてインストール指示に従ってください。他の言語として、モンゴル語とベトナム語のフォントも入ってます。

*またイチゴジャムレシピさんのIchigoJam 1.2 フォント（CC BY）を使用することも可能です。以下のURLでダウンロードができます。

IchigoJam フォント TrueType @

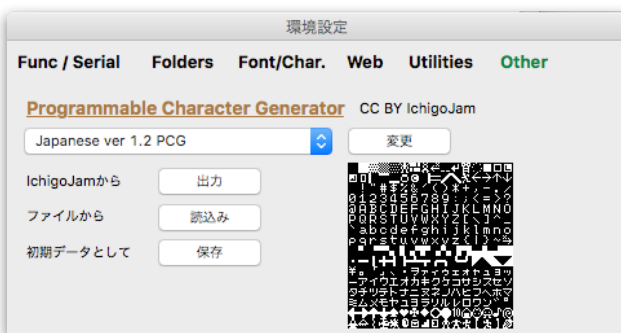
<http://15jamrecipe.jimdo.com/tool/フォント-truetype/>

IchigoLatte用のフォントのインストール

IchigoLatte-for-Displayというフォントを同梱しています。これもあらかじめインストールして下さい。

使用するフォントを設定

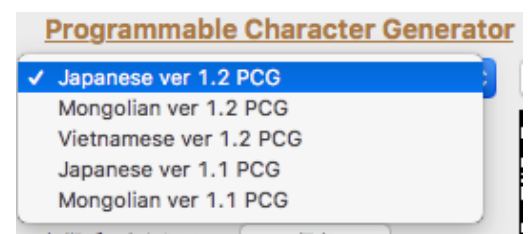
フォントの選択：IJUtilitiesを起動し、**環境設定**の中の**FONT/Char.**を選んで、上記でインストールしたIchigoJam用およびIchigoLatte用のフォントを選択してください。



キャラクターグラフィックの設定：MONITOR表示などで画像文字を使用しています。場合によってはこれが異なったものを利用する場合があり変えることができますが、デフォルトで使っている場合でも異なった画像文字になってしまっている場合があります。

そんな時は、**環**

境設定の中の**Other**で、ポップアップメニューから使用するデータを選んで「**変更**」して下さい。

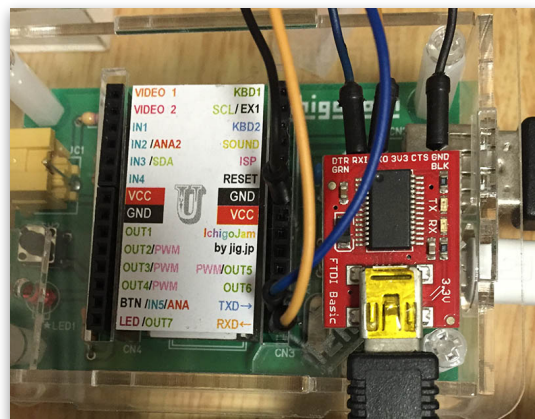


配線

3.3V対応のUSBシリアル変換器が必要になります。またそのドライバソフトのインストールが必要になります。

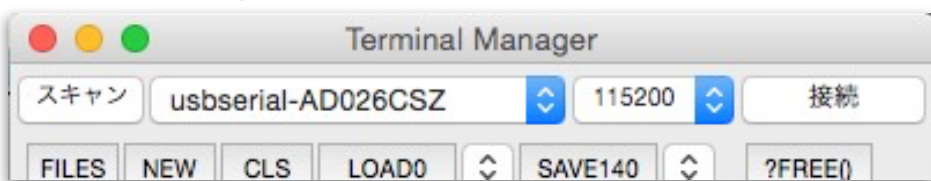
- ・ IchigoJamの**TX**とUSBシリアル変換器の**RX**
- ・ IchigoJamの**RX**とUSBシリアル変換器の**TX**
- ・ IchigoJamの**GND**とUSBシリアル変換器の**GND**
- ・ USBシリアル変換器とコンピュータ

それぞれをつないで使います。

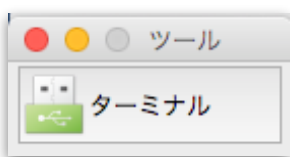


通信

配線をしたのち、ターミナルウインドウの「**スキャン**」ボタンをクリックして、**ポップアップメニュー**から、USBシリアル変換器を選択して、「**接続**」ボタンをクリックします。Windowsの場合は名前ではなく、「COM3」のようにCOM+番号といった表示なので、どれが対象の変換器なのかわかりにくいので注意して下さい。（IchigoJamのデフォルトの通信速度は115200bpsです。）



接続が成功すると、「**接続**」ボタンの名称が「**切断**」に変わります。また、ツールウインドウのいちばん上のターミナルのアイコンの色が「黒」から「**緑**」に変わります。



ウインドウの説明

《ツールウインドウ》

●新規Jamボタン、新規Latteボタン、新規Questボタン

IchigoJam、IchigoLatte、IchigoQuest用の編集ウインドウを開きます。

●開くボタン

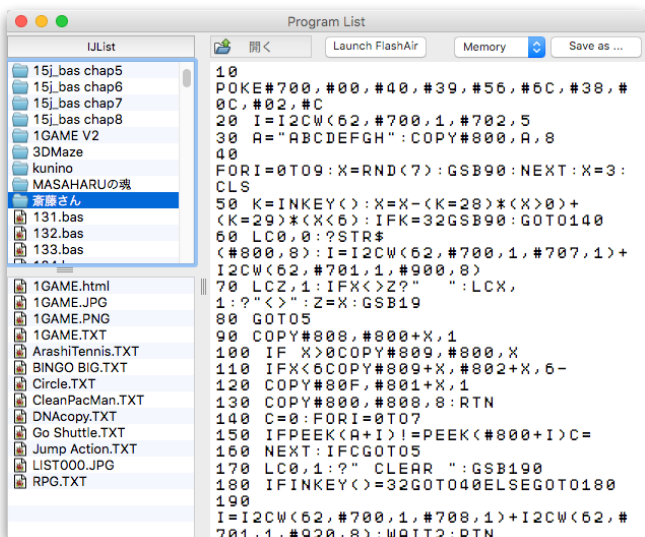
IchigoJamのプログラムまたはIchigoLatteのプログラムを読み込みます。

IchigoLatteのファイルは拡張子が.jsである必要があります。

●リストビューアボタン

プログラムを入れてあるフォルダを指定しておくとそのフォルダ内から簡単に**ファイルを選択して開く**ことができます。フォルダーはダブルクリックで中に入れます。

「..」のダブルクリックで上の階層へ行けます。テキストファイル以外に、HTMLファイル、JPEG、PNG、TIFFファイルも表示されます。.BAS拡張子のファイルはメ

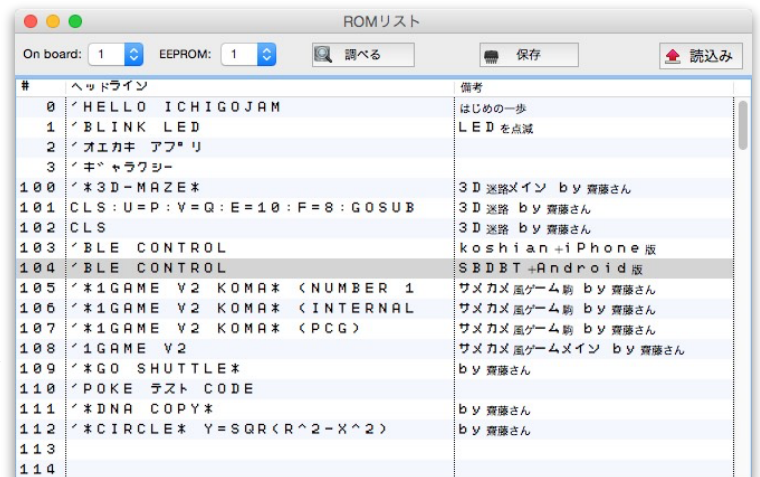


モリー形式として認識します。テキスト形式で認識させる場合は環境設定で.BASの設定を変更してください。

●ROMリスト

IchigoJamに保存されているプログラムや、EEPROMに保存されているプログラムをリストアップし、メモ書きできます。IchigoJam, EEPROM、それぞれ9種類までのデータを保存しておけます。「調べる」のボタンで読み取れます。また、**保存番号を選択して、読み込んだり、IchigoJam上のプログラムを保存**することができます。

●読み込みボタン



IchigoJamに保存されているプログラムまたは、IchigoLatteに繋げたEEPROMに保存されているプログラムを読み込んで、エディタに表示します。ポップアップメニューで読み込むプログラム番号を選択してこのボタンをクリックして下さい。

●ダウンロードボタン

ネットを介してファイルを読み込むことができます。詳しくは後述の「ネットを介した読み書き」をご覧ください。

●Firm書換えボタン（旧lpc21ispボタン）

IchigoJamのファームをアップデートします。lpc21ispというフォルダの中にlpc21isp・lpc21isp.exeが入れてあります。**＊あらかじめ、設定でlpc21ispのファイルを選択しておく必要があります。**接続先はターミナルウィンドウで選択されているデバイスに接続しますので、正しく選択した状態でlpc21ispのボタンをクリックして下さい（現在、正しくないとハングアップするかと思います）。すでに接続済みの状態だった場合は、自動的に切断した後、書き換え作業になりますので、いちいち切断する必要はありません、が、書き換え後、自動で再接続はしない仕様になっています。

- 1、USB-TTL変換モジュールを繋げる
- 2、IchigoJamのISPとGNDをつないだ状態で電源をON
- 3、Firm書換えボタン
- 4、ポートを設定し、Firmのファイルを選択
- 5、スタートボタンを押すとインストールされます。

●マシン語変換ボタン（旧Bin2pokeボタン）

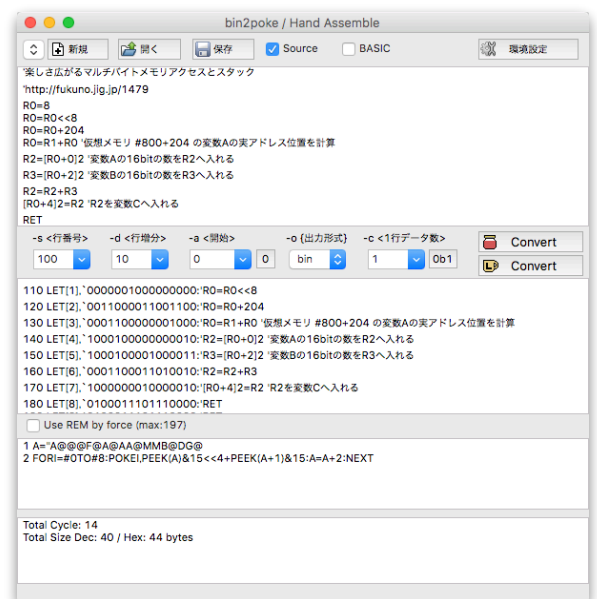
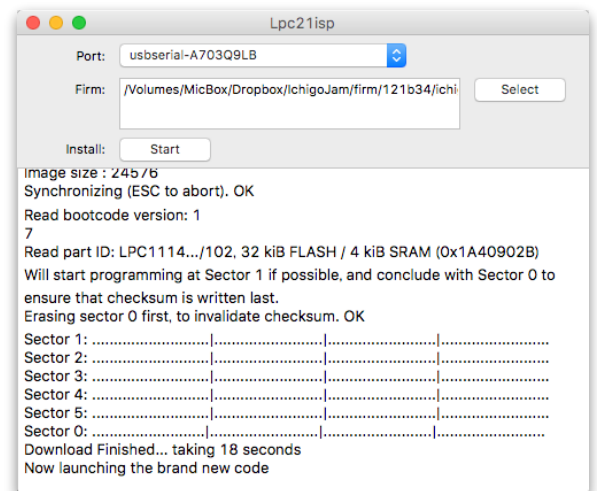
IchigoJamまたはIchigoLatte用マシン語の変換用エディタを開きます。

1、IchigoJam-FANで上がった「Cで書いたusr()で呼び出すマシン語部を、BASICのpokeに変換するツール」（<https://www.facebook.com/groups/ichigojam/permalink/633543440118746/>）をIUtilitiesから使えます。**各種ファイルのインストールと、設定でフォルダを選択しておく必要があります。**必要なファイルの詳細は後述のBin2Pokeの章をご覧ください。

2、ハンドアッセンブルによるマシン語変換もBin2Pokeと同じようにできます。

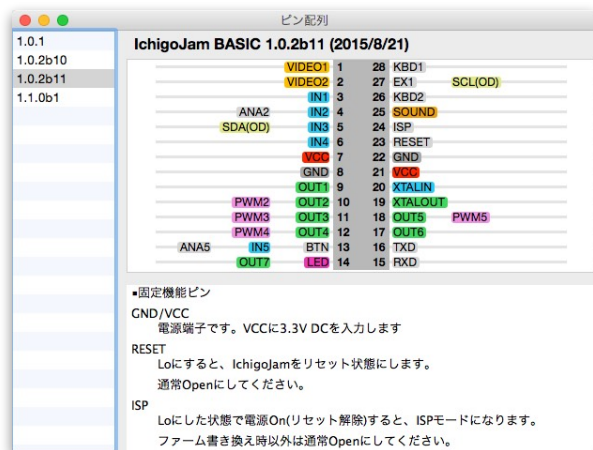
●コマンド一覧ボタン

テキストファイル一覧を表示する機能があります。3つまでのフォルダを指定でき、その中のファイルをリスト表示できます。例えば、イチゴジャムレシピさんのサイトにあるコマンドのファイルを <https://github.com/fu-sen/IchigoJam-BASIC> や <https://github.com/fu-sen/PanCake-COMMAND> でダウンロードしたファイルが入ったフォルダを選択するという使い方があります。左上のポップアップでフォルダを選択できます。



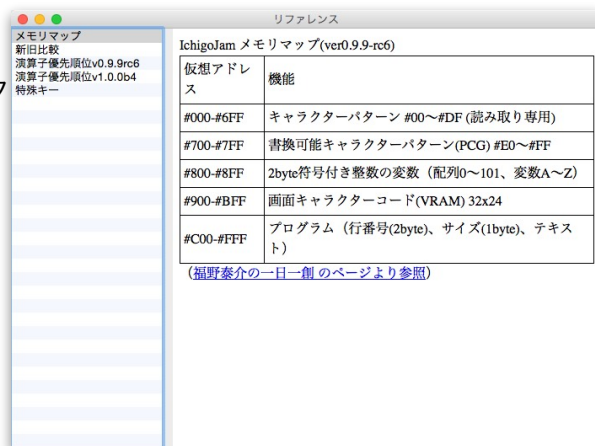
●ピン配列ボタン

サンプルとして、添付したIchigoJam-Pinのフォルダを選択すると読めます。ピン配列は横12個10個のタブ切りです。このデータはIchigoJam-FANで公開されたデータをもとに作成しています。公式なものではないため不正確な部分があるかもしれませんのでご注意ください。



●リファレンスボタン

サンプルとして、添付したIchigoJam-Referのフォルダにデータを作成してあります。このフォルダを選択してご利用ください。ここでの表示はHTMLファイルを表示する仕組みになっています。



●環境設定ボタン

Func/Serial

- 5種類のユーザー設定ボタンの命令を編集登録できます。16進数の場合は&hを頭につけてください。複数の命令をつなぐときは+を使ってください。つまり、文字列として&hと+は使えません。

- 通信速度以外のシリアル設定ができます。IchigoJamで使用する場合は基本的にはデフォルトの設定のままで大丈夫です。



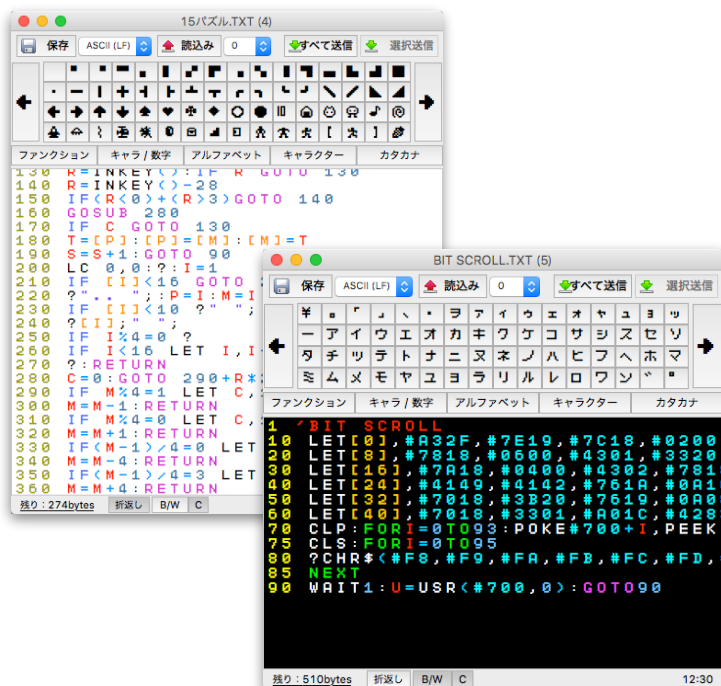
Folder

コマンド一覧（3種類まで）、PIN配列、リファレンス、プログラムデータ、ROMリスト用の5種類のフォルダを選択してしておくことができます。



Font/Char.

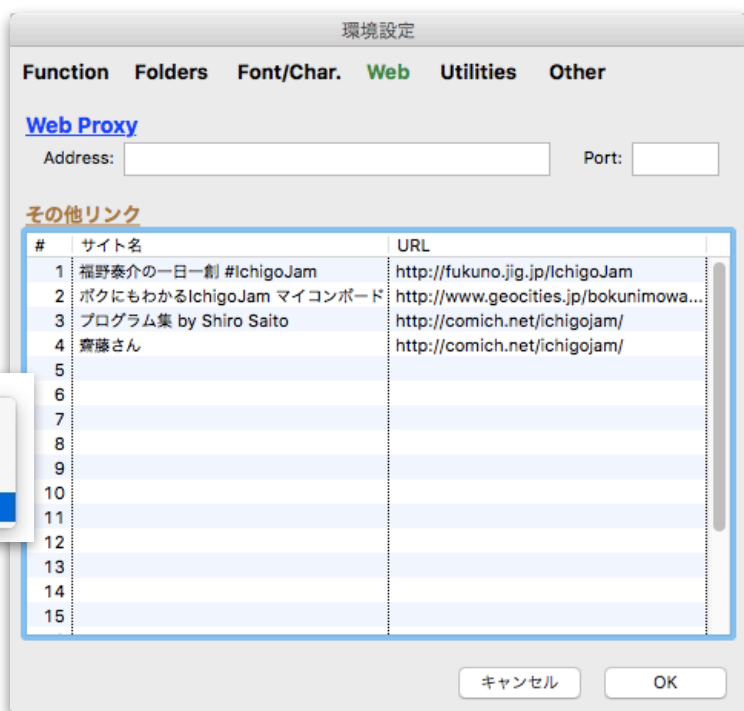
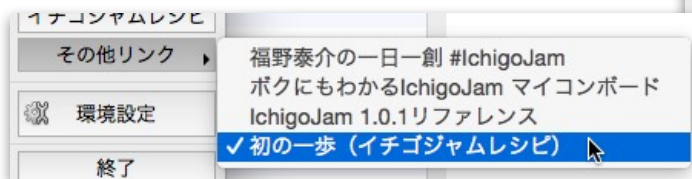
●IchigoJam用のフォント、IchigoLatte用のフォントおよび、どちらをメインとして使うのかを選択して下さい。IchigoJam用のモンゴル語やベトナム語のフォントも選択することができます。また、ターミナルセンター、エディター、プリント時の文字サイズを指定します。エディタウインドウは一度閉じないと、文字のサイズは反映されません。エディターを開いた時の背景色、シンタックスの色を使うかを設定できます（マック版のみの機能）



Web

●MixJuiceでインターネット接続をしますが、プロキシが必要な場合に設定ください。

●20個までURLを登録しておけます。ツールウインドウからご利用ください。



Utilities

lpc21isp

ファームを焼くためのlpc21ispのファイルを指定します。

bin2poke

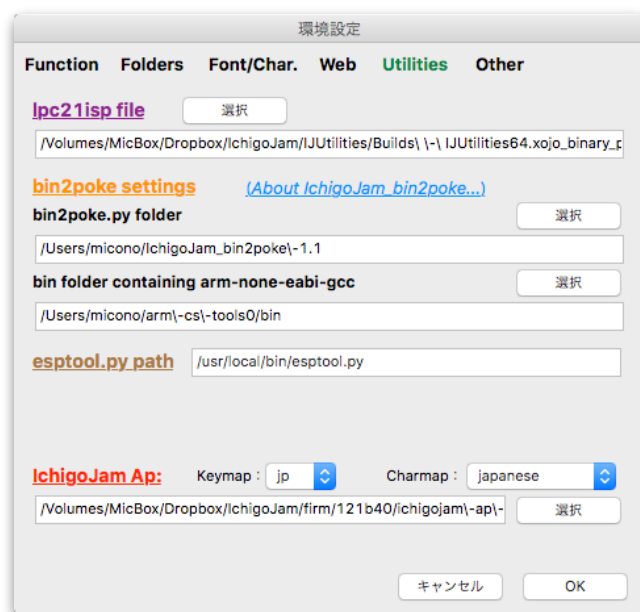
IchigoJamのマシン語を使ったプログラミングでCで書いたプログラムをマシン語に変換するための設定

EspTool

MixJuiceなどESP-WROOM-02にファームを焼くためのソフトなどの指定

IchigoJam Ap

IchigoJam Apのファイルの指定やキーボードタイプの設定などを行います。



Other

Programable Character Generator

キャラクター入力で用いるデータをIchigoJamから読み込んでファイルとして保存できますので、それを読み込んで使ってください。MONITOR表示機能を使う場合、ここで設定されているフォント画像を使います。デフォルトのフォント画像にする場合は、ポップアップから目的のタイプを選んで、変更ボタンをクリックして下さい。

Send Buffer

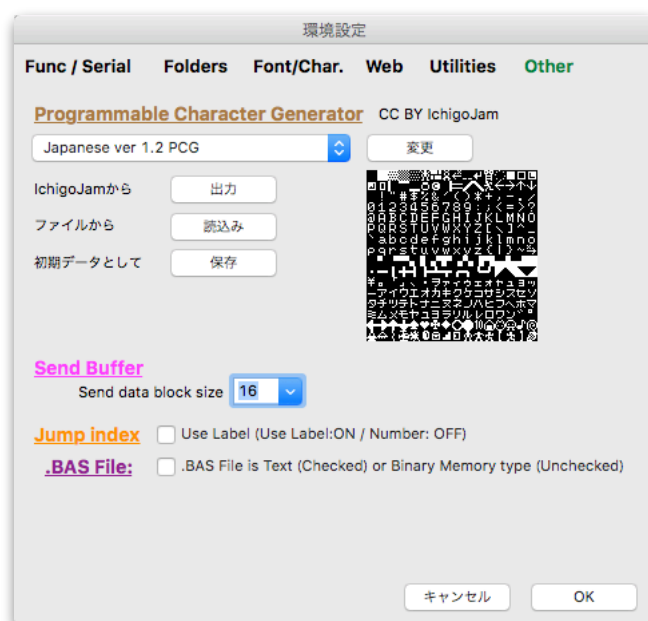
データ転送時の1回の転送のバイト数を設定できます。大きすぎるとデータの取りこぼしをしてしまうので各自の環境で適切な値に設定して下さい。

Jump Index

エディタウインドウで、DENUM/RENUMの際、ラベル対応をデフォルトにするか、ラベル非対応の行番号の置き換えにするかを選択します。

.BAS File

拡張子.BASのファイルをメモリー形式かテキスト形式かどちらの形式で開くのかを設定します。



《ターミナルウインドウ》

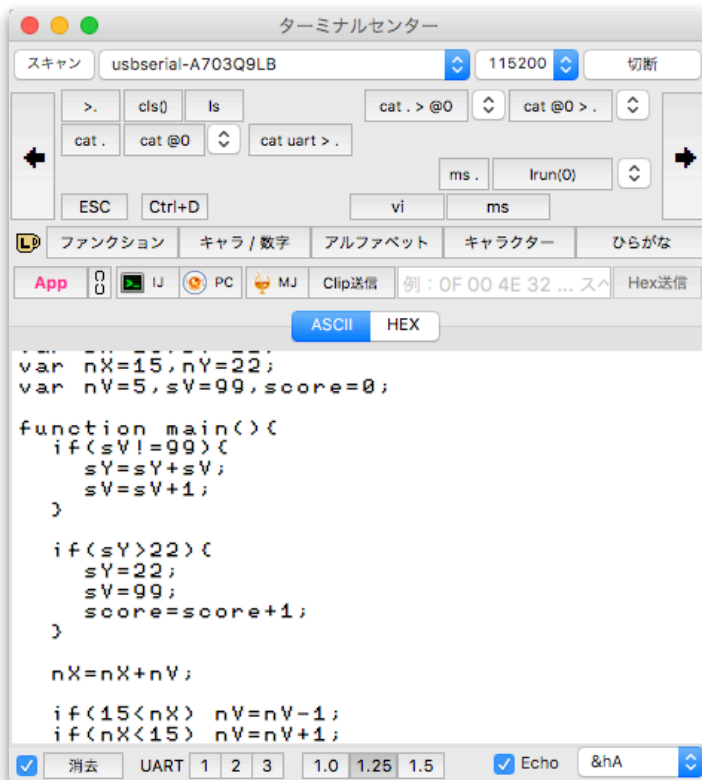
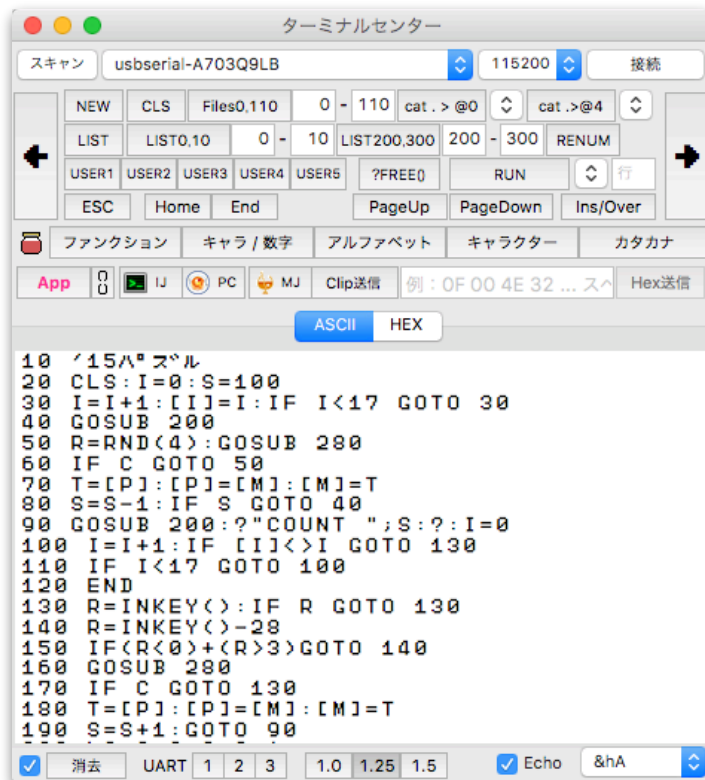
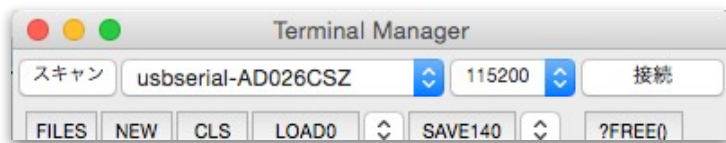
IchigoJam用とIchigoLatte用のウインドウの切り替えができます。ウインドウ左中央のメイソンジャーまたはマグカップのアイコンをクリックすると、その切り替えができます。



●シリアル接続

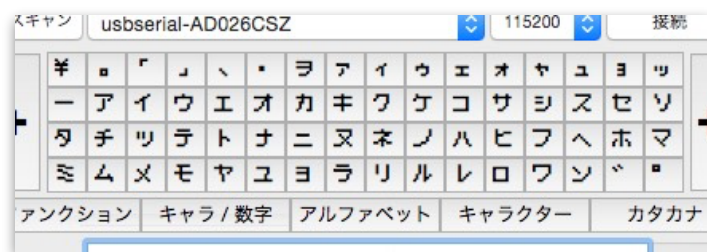
目的のポートを選択して「接続」ボタンで接続できます。目的のポートがどれかわからない場合は、一度、UART変換器を外して「スキャン」

ボタンをクリックし、そこで表示されているポート名を記録した後に、変換器を接続後、もう一度、「スキャン」ボタンを押して新たに現れたものが目的のポートであると思います。



●ファンクション・キャラ入力ボタン

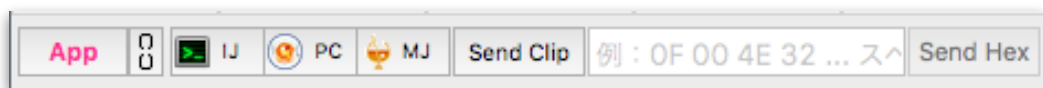
「ファンクション」「キャラクター」「アルファベット」などのボタンをクリックすると、いろいろなボタンが表示されます。よく使うコマンドや、キーボードから入力しにくい文字は、このパネルを使うとよいと思います。



●App & Link ボタン

Appボタンをクリックすると、UARTに対する操作が

らIchigoJam Ap版に対する操作に変わります（初回はIchigoJam Apの場所が聞かれます）。「App」ボタンの右のボタンをオンにすると、UARTからのデータとAppからのデータを双方に送り合います。IchigoJamどうして通信しあうようなプログラムのデモ的な使い方ができるかもしれません。注意）Windows版の場合はIchigoJam Apを起動できるだけです。

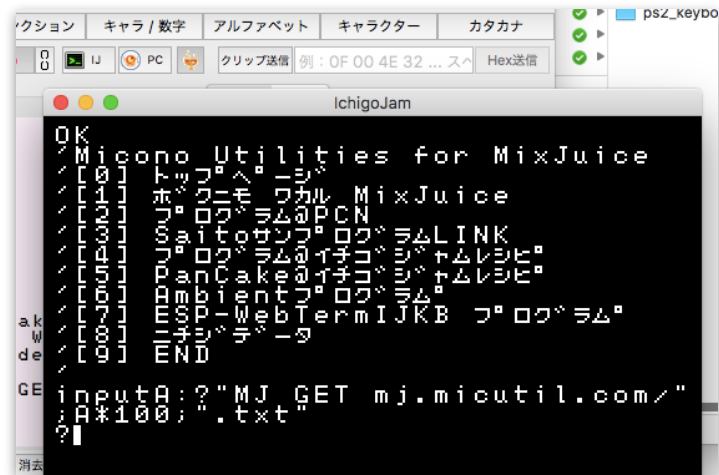
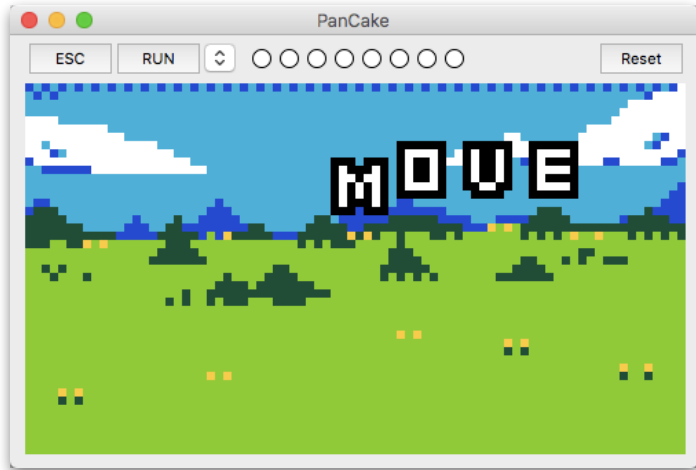
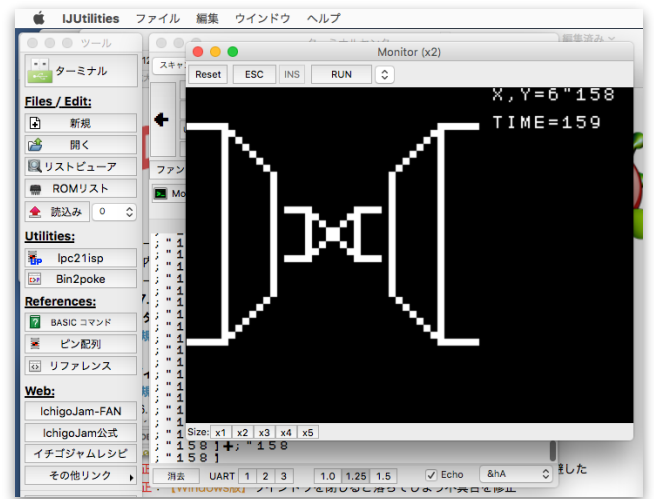


●IchigoJam Monitorボタン

IchigoJamの画面表示ウインドウを表示します。この機能に関しては、別途説明があります。

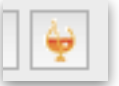
●PanCake シミュレーター

PanCakeシミュレータです。IchigoJam またはIchigoJam Apで動かしたPanCake プログラムをパソコンで動かします。サウンド関連のコマンドは未対応です。



●MixJuice 機能ボタン

IchigoJamまたはIchigoJam Apで動かしたMixJuiceコマンドを実行します。



●クリップ送信ボタン

コピーされている文字を送信します。

●16進入力フィールド・Hex送信

16進数でキャラクターを入力できます。半角英数字で2文字ずつ入力して、「Hex送信」ボタンをクリックすると、対応するキャラクタが送信されます。

●ターミナルフィールド

キーボードから入力できる英数字はここで入力すれば通信できます。このエリアにプログラムファイルをドラッグ&ドロップすることもできます。

*文字のペーストは現在未対応です。

●消去ボタン左のチェックボックス

受送信したシリアルデータのASCII・HEX処理をオン・オフします。

●消去ボタン受送信

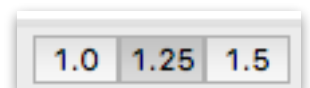
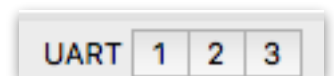
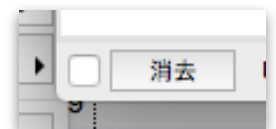
受送信したASCII・HEXデータ表示を消去します。

●UART1,2,3

UARTのモードを変更します。

●行間1.0/1.25/1.5

行間1.25がデフォルト。1.0は行間なし

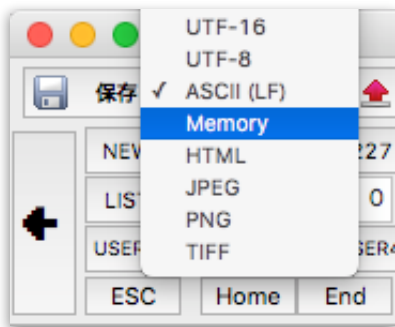


《エディタウインドウ》

IchigoJam、IchigoLatte、IchigonQuestのプログラムを編集するためのウインドウです。ツールウインドウの「開く」「リストビューア」、「インポート」などでプログラムを読み込むことができます。

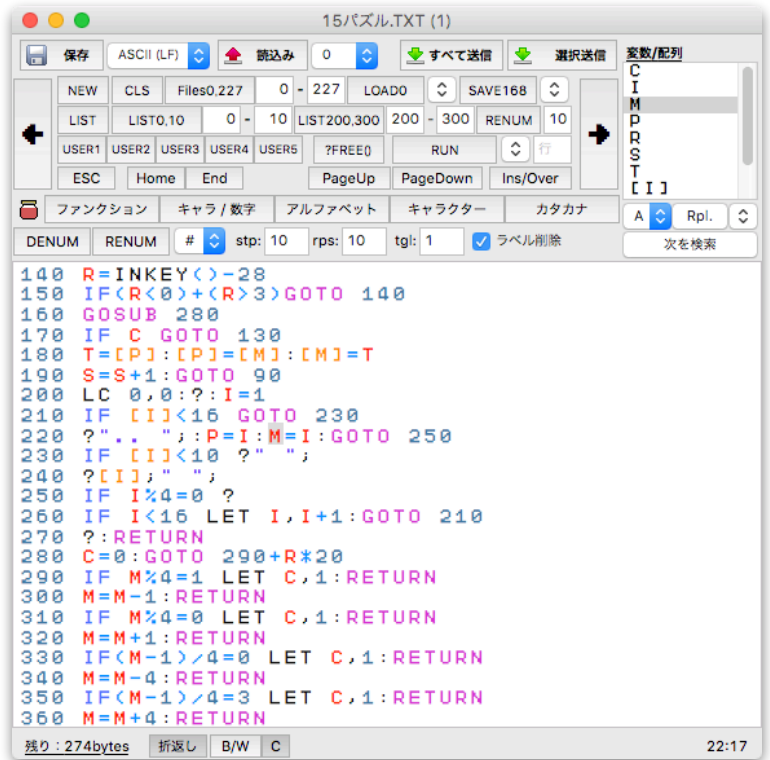
●保存・出力ボタン

編集したプログラムを保存します。文字コードを選択して保存したり、ホームページ用（HTML用）、JPEG、PNG、TIFFなどの画像にも出力ができます。またMemory形式で保存した場合JamToastで利用出来るファイル形式になります。



IchigoLatteのプログラムを保存

は現在、IchigoLatteには対応していません。



存した場合、拡張子は.jsになります。Memory形式での

保存および保存形式の右側に「オンライン」のチェックボックスがあります。これにチェックを入れるとネットを介してファイルをアップロードできます。詳しくは後述の「ネットを介した読み書き」をお読みください。



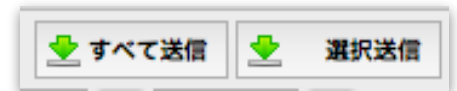
●読み込みボタン

IchigoJamまたはIchigoLatteからプログラムを読み込みます。ボタンの右のポップアップメニューで対象のプログラム番号を選んでから、読み込みボタンをクリックして下さい。現在編集中のプログラムに上書きするので、新規に別を開く場合はツールウインドウから行ってください。



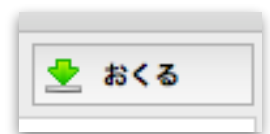
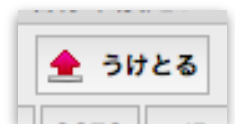
●すべてを送信、選択送信ボタン

プログラムをIchigoJamまたはIchigoLatteに転送します。一部分を送信する場合は、送信したいプログラム部分を選択して、選択送信ボタンをクリックしてください（選択送信はIchigoLatte未対応です）。



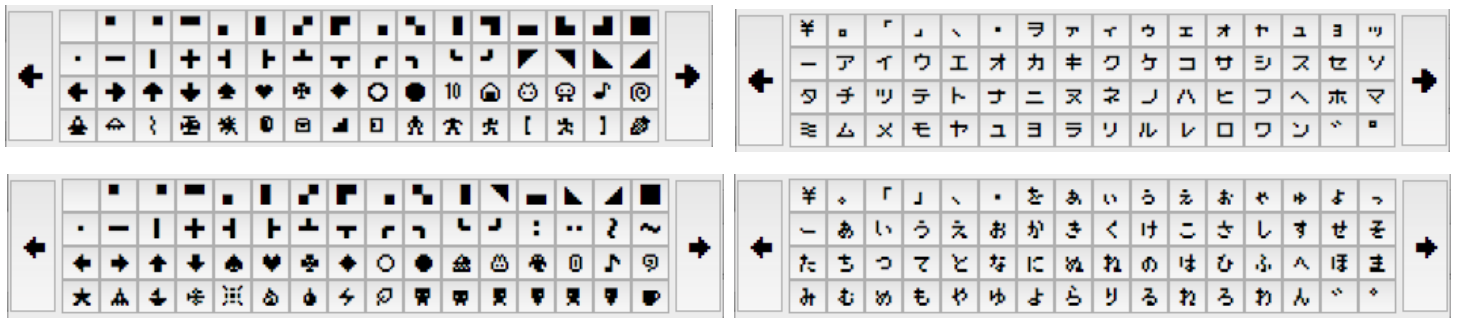
●IchigonQuestの場合の読み込み（うけとり）と送信（おくる）ボタン

- ・IJUtilitiesにプログラムを送る場合は、まず、IJUtilities側の「うけとる」のボタンをクリックした後に、IchigonQuest側の「つうしん」>「おくる」の操作をします。
- ・IchigonQuestにプログラムを送る場合は、IchigonQuest側の「つうしん」>「うけとる」を選択した後、IJUtilities側の「おくる」のボタンをクリックします。



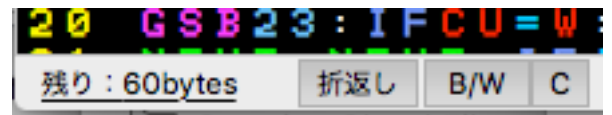
●特殊な文字の入力

キャラクターや、カタカナ・ひらがななどキーボードからの入力が必要な文字の入力は、「キャラクター」「カタカナ・ひらがな」のボタンをクリックして表示される、文字リストから簡単に入力できます。



●プリント

編集したプログラムを印刷できます。ファイルメニューの「プリント…」から行ってください。



●Uppercase / Lowercase

リスト全体（選択範囲のみではない）に対して大文字・小文字の変換を行います。コメント内・引用符内の文字に対しては大文字・小文字変換は行われません。

●残りバイト数の表示

*IchigoLatte未対応

●折返し表示

●白黒反転表示 (BW)

●シンタックスカラー (C)

*Windows未対応

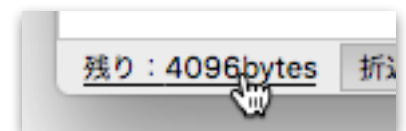


●ファンクションの「RENUM」

ステップ行番号を指定して編集中のプログラムの行番号の振り直しができます。GOTO,GOSUBの番号も合わせて変更されます。*IchigoLatte未対応

●プログラム残量表示の切り替え

プログラム残量表示の部分をクリックすると1024と4096が切り替わります。LPC1115をお使いの場合などは、クリックして切り替えて下さい。



●変数、配列の一覧と変数の検索置換

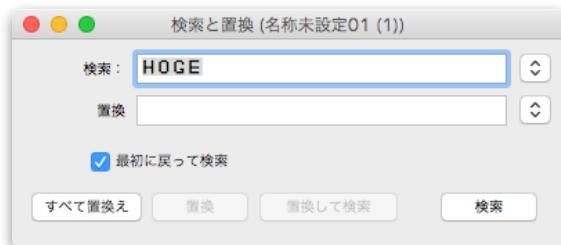
(IchigoJamのみ、IchigoLatte未対応)

プログラムで使用している変数や配列の一覧が、エディタ左上に表示されます。その変数を選択して検索したり、置換することができます。



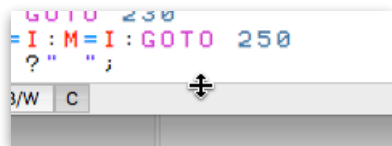
●一般的な検索置換

CommandキーまたはALTキー+F、または、編集メニューの「検索と置換」で普通の検索、置換ができます。



●プログラム内容のリスト表示

プログラムの内容を簡単なリストで見ることができます。またリストを選択すると、その部分が表示されます。エディットウィンドウの下の方から引き出して表示して下さい。



type	Num	Data
EC	00050	17
PAGE	00052	
OMP	00053	GOTO
PAGE	00057	
EC	00058	30
OL	00060	<-- End of Line -->
EC	00061	40
PAGE	00063	
OMP	00064	GOSUB
PAGE	00069	

●micro:bit Javascriptとしてコピー

IchigoJamのプログラムをmicro:bit Javascriptに変換してコピーすることができます。

右クリックすると右の図の様なメニューが表示されます。「micro:bit JSとしてコピー」を選択すると、micro:bit Javascriptに変換したプログラムがコピーされているので、micro:bitのサイトのJavascriptにペーストして下さい。



【対応】

●コマンド

ABS, ANA, BEEP, CHR\$, CLS, FOR/TO/NEXT, GOSUB/GSB/RETURN/RTN, IF/THEN/ELSE, IN, LED1/LED0, LOCATE/LC, OUT, PRINT/?, RND, TEMPO, WAITほか

●関数・イベントは@のラベルで表し、最後はRTNで挟む

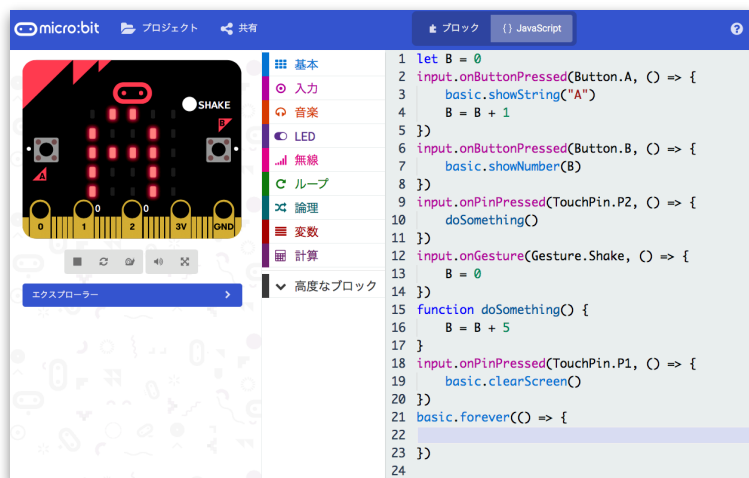
ずっと：@forever

ボタン：@ButtonA、@ButtonB

端子：@TouchP0,@TouchP1,@TouchP2

ゆさぶられた：@Shake

関数：@関数名



コマンドの使い方については、「[micro:bit用のプログラムのコマンドリファレンス](#)」を参考にして下さい。「Examples」の中の「for microbit」の中にサンプルがありますので、参考にして下さい。

【DENUM/RENUM機能】

行番号の割り付けは結構めんどくさいものです。このナンバリング機能は、従来のRENUM機能をより便利にしたものです。

1、行番号を書かずにプログラムを書く

DENUMボタンをクリックすると、自動で行番号が削除され、GOTO, GOSUBのジャンプ先行番号がラベル化され、ジャンプ先の行のところに対応する名称のラベル名が挿入されます。GOTO/GOSUBのジャンプ先をラベルで表記できるので、行番号なしでプログラムを書くことができます。

2、行番号を気にしないでプログラムを書く

行番号があろうがなかろうが、行番号が前後してようが、ここRENUMボタンをクリックすると、最初から順に行番号を振ってくれます（ジャンプ先の行番号やラベル名は正しい必要があります）。次の4種類のオプションがあります。



- ・ RENUM右のポップアップ：#が行番号対応、@がラベル対応
- ・ stp: 行番号のステップ（最小値1）
- ・ rps: 最初の行番号（最小値1）
- ・ tgl: 対象の行番号（最小値0か1、rps>tglの必要がある）
- ・ Del Label Line: ラベル行を削除する

例えば、

stp=3, rps=2の場合、行番号が2,5,8,11…となる

stp=10, rps=50, tgl=13の場合、13行め以降を50番からナンバリングする（図）

サブルーチンごとに行番号を切りの良い番号から付けたい時などに利用できます

```
10 I=I+1:IF [I]<>I GOTO 13
11 IF I<17 GOTO 10
12 END
13 R=INKEY<>:IF R GOTO 13
14 R=INKEY<>-28
15 IF<R<0>+<R>3>GOTO 14
16 GOSUB 28
17 IF C GOTO 13
```

```
10 I=I+1:IF [I]<>I GOTO 50
11 IF I<17 GOTO 10
12 END
50 R=INKEY<>:IF R GOTO 50
51 R=INKEY<>-28
52 IF<R<0>+<R>3>GOTO 51
53 GOSUB 65
54 IF C GOTO 50
```

注意：

- ・ ラベル名は@から始まる英数字をラベルとして認識します。
- ・ ラベルは行頭にある必要があります。
- ・ DENUMの変換でラベル名は@LV行番号となっていますので、RENUM・DENUMを繰り返したことによって、違う行なのに同じラベル名になってしまうことのないように注意が必要です。

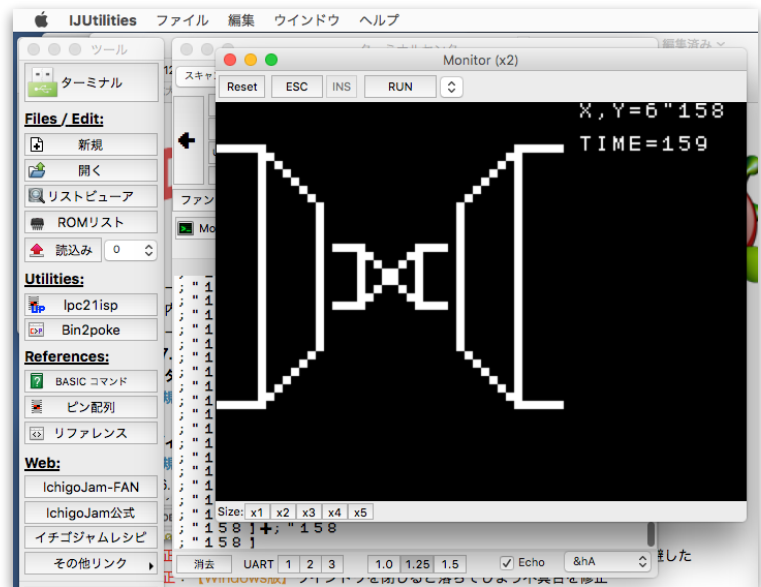
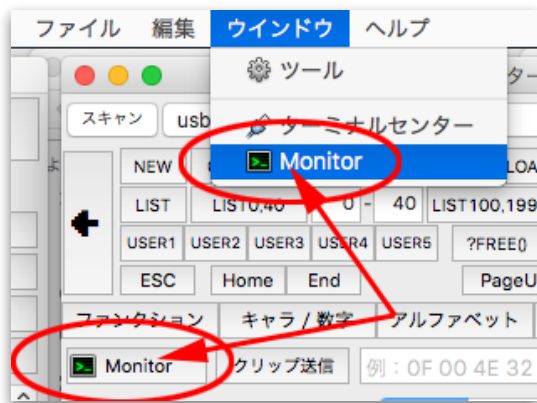
【IchigoJamモニター機能】

シリアルでつないでいるIchigoJam表示状態をパソコンの画面でシミュレーションします。以下に従ってご利用ください。右の画像は、3D迷路（SHIROのモバイル日記より）をプレイ状態をスクリーンショットしたものです。

1、ウィンドウの表示

ターミナルセンターのウィンドウにある

「Monitor」ボタンまたは、ウィンドウメニューの「Monitor」を選択することで、モニターウィンドウを表示させることができます。

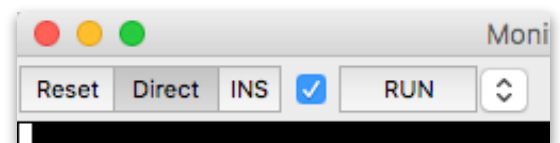


2、IchigoJamと接続

ターミナルウィンドウの「接続」ボタンでIchigoJamと接続して下さい。

3、画面のリセット（Resetボタン）

IchigoJamの画面状態とモニターウィンドウの表示を同じにするために、まず「Reset」ボタンをクリックして下さい。画面が消去されます。

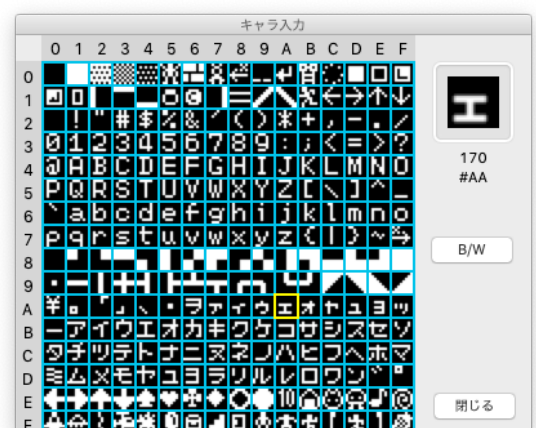


4、挿入と上書き（INSボタン）

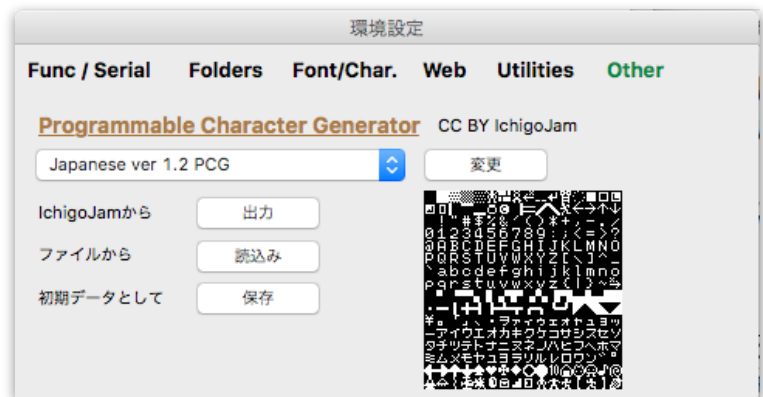
ダイレクトモードで、文字の入力を挿入と上書きのモードを切り替えます。デフォルトは挿入モードですが、他の方法で挿入・上書きを切り替えているかもしれませんので、実際の状態はパソコンの画面の状態とは異なっている場合も考えられますのでご注意ください。

5、文字やキャラクターの入力

ターミナルウィンドウか、モニターウィンドウで入力した文字が反映されます。特殊なキャラクター・カタカナなどは、ターミナルウィンドウのボタンから入力して下さい。また、モニターウィンドウ内をクリックすると、右のような文字入力用のウィンドウが開きます。ここから選択も可能です。



キャラクターを作り変えている場合は、キャラクターを作り変えた状態で環境設定のOtherで、キャラクターを「出力」した後、そのファイルを「読み込み」することで表示可能になります。



6、実行と停止 (RUN/LRUNボタンとESCボタン)

モニターウインドウまたはターミナルウインドウのRUN・LRUNボタンでプログラムを実行することができます。またESCボタンでプログラムを停止できます。手入力でのRUNなどすると実行モードにならないかもしれませんのでご注意ください。



＊動作はまったく同じようにならない場合もありますのでご了承ください。すべての状況に対応させることは困難ですが、不具合を示すプログラムを教えていただくと改良可能かもしれませんのでご協力ください。

＊「RUN」ボタンの左のチェックボックスは、BTN(***)のコマンドを使っている場合のキーダウン・アップの2回送信に対応させるためのチェックボックスです。実際にキーダウン・アップを認識して送信しているのではなく単純に2回送信しています。

7、表示サイズ

IchigoJam 1.2.0-big-beta (<https://www.facebook.com/groups/ichigojam/728680847271671/>) の**16x12のBig表示**と、

IchigoJam 1.2.0 (正式版 :

<http://ichigojam.net/farm.html>) にある**16x6の**

AQM1248A LCD表示に対応しています。その他、

VIDEO1～8までにも対応し

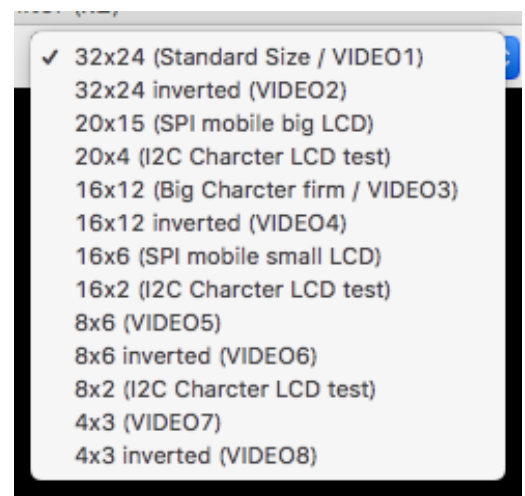
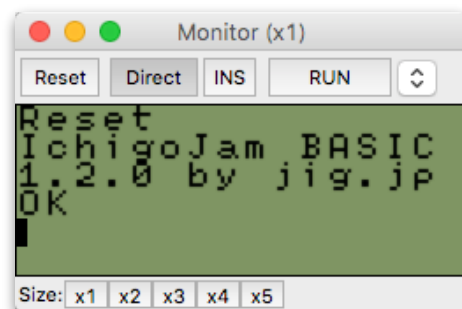
ています。なお表示サイズは

自動的に認識はされませんの

で、ポップアップメニューか

ら、対応したサイズを選択して下

さい。



【IchigoJam Ap版・PanCake, MixJuiceのシミュレータの連携】


IchigoJam本体・Ap版とPanCake本体・シミュレータ／MixJuice本体・シミュレータを連携させて動かすことが可能です。

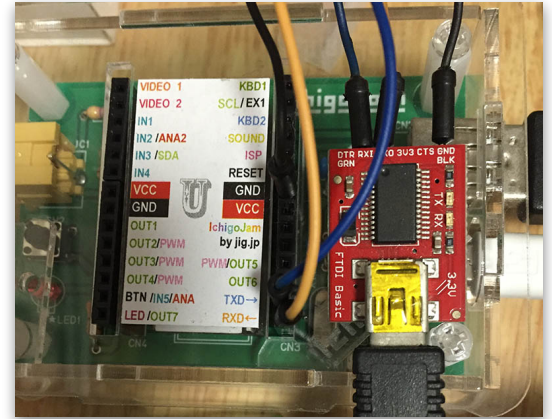
①IchigoJam本体とPanCakeまたはMixJuiceのシミュレータを使う場合 (対応：Mac版、Win版、Linux版、RasPi版)

接続


- ・ IchigoJamの**TX**とUSBシリアル変換器の**RX**
- ・ IchigoJamの**RX**とUSBシリアル変換器の**TX**
- ・ IchigoJamの**GND**とUSBシリアル変換器の**GND**

●IchigoJam**本体** → PanCake**シム**

ターミナルウインドウの  ボタンをクリックしPanCakeシミュレータウインドウを表示させた状態で、PanCake用に作られたプログラムを実行してください。



●IchigoJam**本体** ⇄ MixJuice**シム**

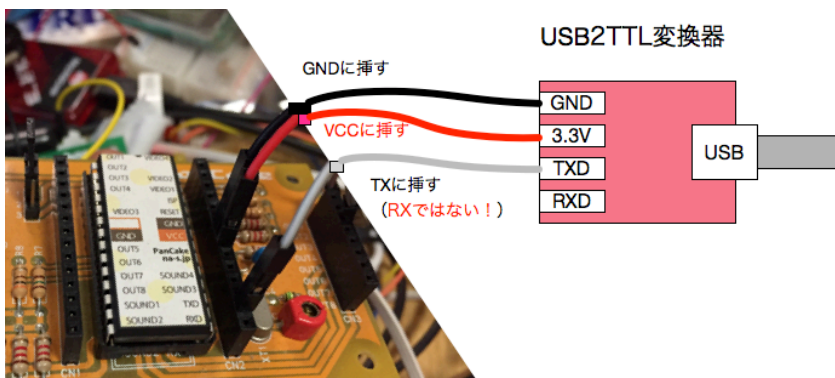
ターミナルウインドウの  ボタンをクリックした状態になっていると、MixJuiceのコマンドの処理を行い、その結果をIchigoJamに送信します。


②IchigoJam Ap版とPanCakeまたはMixJuiceの本体を使う場合 (対応：Mac版)

Windows版のIchigoJam Apに対応していません。また、IchigoJam Ap版のLinux, RasPi版がないため、Mac版のみの対応になります。

●IchigoJam **Ap版** → PanCake**本体**

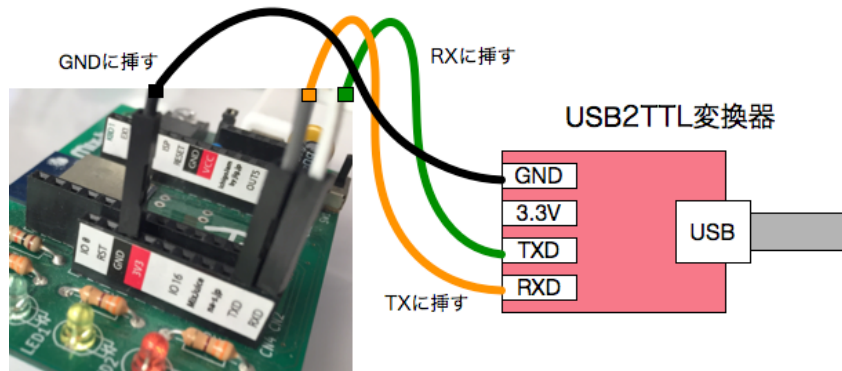
接続：PanCake単体（IchigoJamはつけないで）、USBシリアルとPanCakeのGND同士と、USBシリアルのTXとPanCakeのTXを繋げて下さい。また以下の図では電源をUSBシリアル変換器から供給する図になっていますが、電源が足りない場合は他から供給して下さい。




ターミナルウインドウのAppボタンとその隣のリンクボタン  をクリックし、IchigoJam Ap版でPanCake用に作られたプログラムやコマンドを実行してください。

●IchigoJam Ap版 ⇔ MixJuice本体

接続：MixJuice単体（IchigoJamはつけないで）、USBシリアルとMixJuiceのGND同士と、USBシリアルのRXとMixJuiceのCN4のTXを、USBシリアルのTXとMixJuiceのCN4のRXを繋げて下さい。電源はUSBシリアルから取るのではなく、CN1から供給するのが無難です。




ターミナルウィンドウのAppボタンとその隣のリンクボタン  をクリックし、IchigoJam Ap版でMixJuice用に作られたプログラムやコマンドを実行してください。


③IchigoJam Ap版とPanCakeまたはMixJuiceのシミュレータを使う場合 (対応：Mac版)

Windows版のIchigoJam Apに対応していません。また、IchigoJam Ap版のLinux, RaspPi版がないため、Mac版のみの対応になります。またPanCakeもMixJuiceもシミュレータなので、本体なしで動作を実行できます。

●IchigoJam Ap版 → PanCakeシム

 をクリックしてIchigoJam Apを起動、 をクリックして、PanCakeシミュレータを表示させた状態で、PanCake用に作られたプログラムやコマンドを実行してください。

●IchigoJam Ap版 ⇔ MixJuiceシム

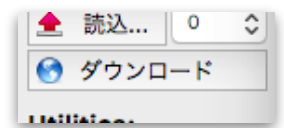
 をクリックしてIchigoJam Apを起動、 をクリックして、PanCakeシミュレータを表示させた状態で、PanCake用に作られたプログラムやコマンドを実行してください。

【ネットを介したファイルの読み書き】

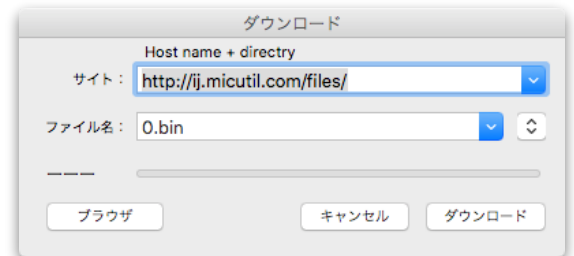
(扱うファイルのフルパスに日本語が含まれる場合は使用できません)

●ダウンロード（読み込み）

- 1、Toolウインドウに「ダウンロード」というボタンがあります。これをクリックするとダウンロード先を設定するダイアログが開きます。
- 2、「サイト」にダウンロード先のURLとダウンロードするファイルが入ったパスを入力します。
- 3、「ファイル名」にダウンロードするファイルのファイル名を入力して下さい。



例えば、FlashAirからダウンロードする場合は、サイト名は「http://flashair/」となり、ダウンロードしたいファイルが入ったフォルダが「files」ならば、「http://flashair/files/」となります。



Kidspodからのダウンロードの場合、例えば図の場合は、Mixjuice GET URLのところを参考にサイト名が「http://kidspod.club/mj」、ファイル名は15となります。

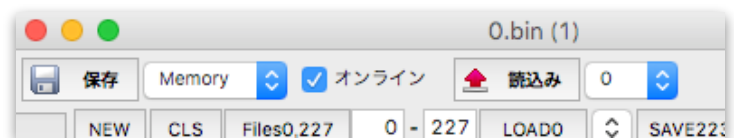


- 4、FlashAirからダウンロードする場合は、Wi-Fiの接続先をFlashAirのWi-Fiに切り替えて下さい。

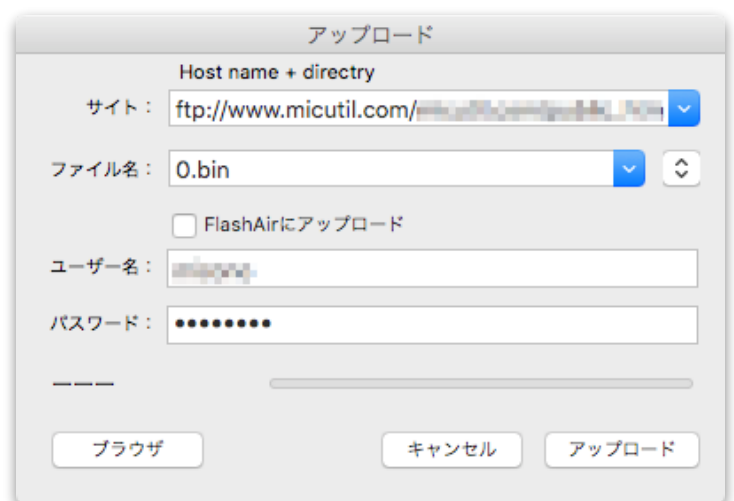
- 5、「ダウンロード」ボタンをクリックするとファイルをダウンロードして、表示されます。

●アップロード（書き込み）

- 1、Editウインドウの「オンライン」のチェックボックスにチェックを入れます。
 - 2、ファイルタイプを選択して「保存」ボタンをクリックします。
 - 3、「サイト」にダウンロード先のURLとダウンロードするファイルが入ったパスを入力します。
- FlashAirのサイトは「http://flashair/」であるため、ダウンロードしたいファイルが入ったフォルダが「files」ならば、「http://flashair/files/」となります。
- 3、「ファイル名」にダウンロードするファイルのファイル名を入力して下さい。
 - 4、FTPサイトへアップロードする場合は、ユーザーとパスワードも記入して下さい。FlashAirの場合は必要ありません。



- 5、FlashAirからダウンロードする場合は、Wi-Fiの接続先をFlashAirのWi-Fiに切り替えて下さい。
- 6、「アップロード」ボタンをクリックするとファイルが保付されます。



FlashAirに書き込む場合（重要）

FlashAirにファイルをアップロードする場合は、FlashAirの中に、SD_WLANフォルダ（注意、隠しファイルになっています）の中のCONFIGファイル（隠しファイルです）をエディタで開き、UPLOAD=1と書き加えてください。

●サイトの初期設定

環境設定のWebの項目で、プリセットができます。HTTPの欄にダウンロード先を、FTPにアップロード先のURL、ユーザー名、パスワードを入力しておくとは便利です。現在ダウンロード先はFTPに対応していませんが、今後対応予定です。

環境設定

Func / Serial Folders Font/Char. **Web** Utilities Other

Web Proxy

Address: Port:

その他リンク

#	サイト名	URL
1	福野泰介の一日一創 #IchigoJam	http://fukuno.jig.jp/IchigoJam
2	ボクにもわかるIchigoJam マイコンボード	http://www.geocities.jp/bookunimowa...
3	プログラム集 by Shiro Saito	http://comich.net/ichigojam/
4	斎藤さん	http://comich.net/ichigojam/
5	IchigoLatteを楽しもう	http://www.openspc2.org/reibun/Ichi...
6		
7		

ダウンロードURL

HTTP	FTP (Host+DIR)	FTP (Username)	FTP (Password)
http://ij.micutil.co...	ftp://www.micutil....		
http://flashair/files/			

キャンセル OK

【FlashAirファイルツール】

FlashAirにファイルを無線でアップロード・ダウンロードするためのツールを同梱しています。細かな設定や使い方はFlashAirのサイトもしくは同梱のreadme.txtファイルをご覧ください。

同梱のFlashAir_Toolの中にあるSD_WLANフォルダの中のファイルをFlashAirの中のSD_WLANフォルダ（注意、隠しファイルになってます）の中にコピーしてください。

また、ファイルのアップロードを行いますので、CONFIGファイル（隠しファイルです）にUPLOAD=1を書き加える必要があります。

JamToastでFlashAirを使うと無線でファイルをアップロードできるので非常に便利です。

JamToastで読み書きするファイル形式はメモリー形式になりますので、またファイル名がIJ100.BASからIJ227.BASとなりますので、それに合わせたファイル形式、ファイル名で保存して下さい。

アップロード

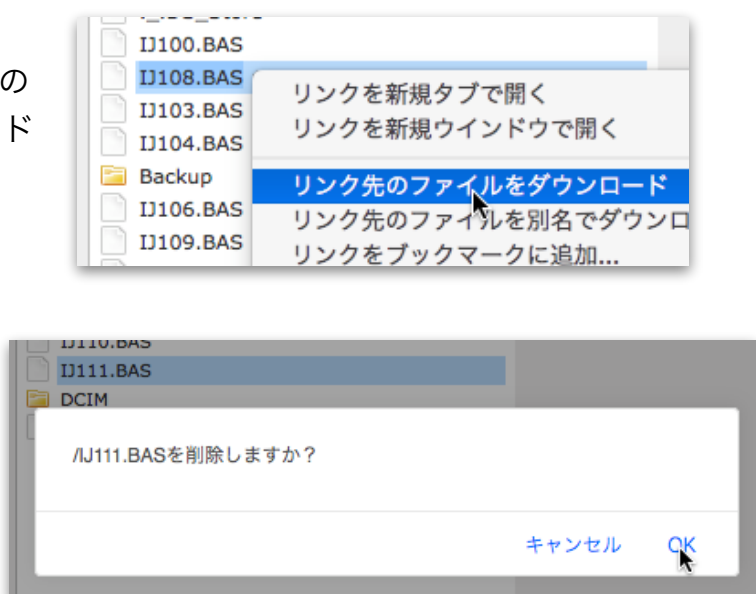
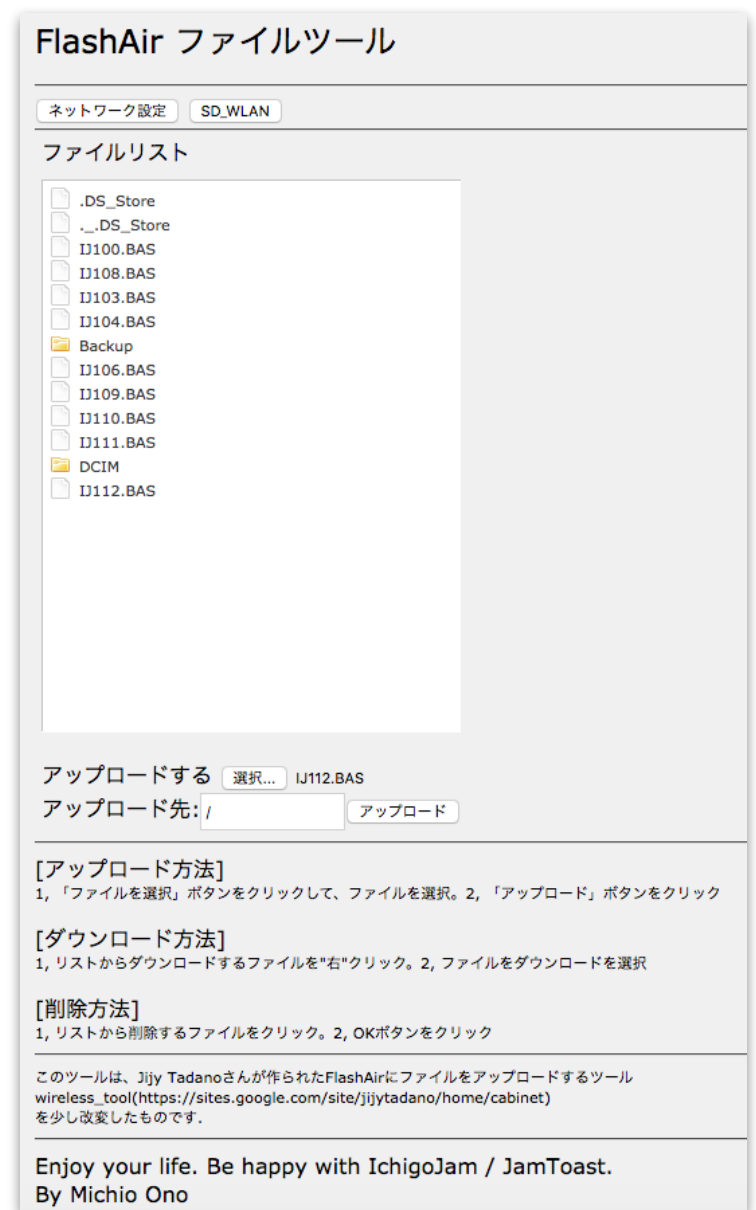
ダウンロードする「ファイルを選択」ボタンをクリックしてファイルを選択したのち、「アップロード」ボタンをクリックして下さい。

ダウンロード

ダウンロードするファイルを右クリック、メニューの中からダウンロードする項目を選んでダウンロードしてください。

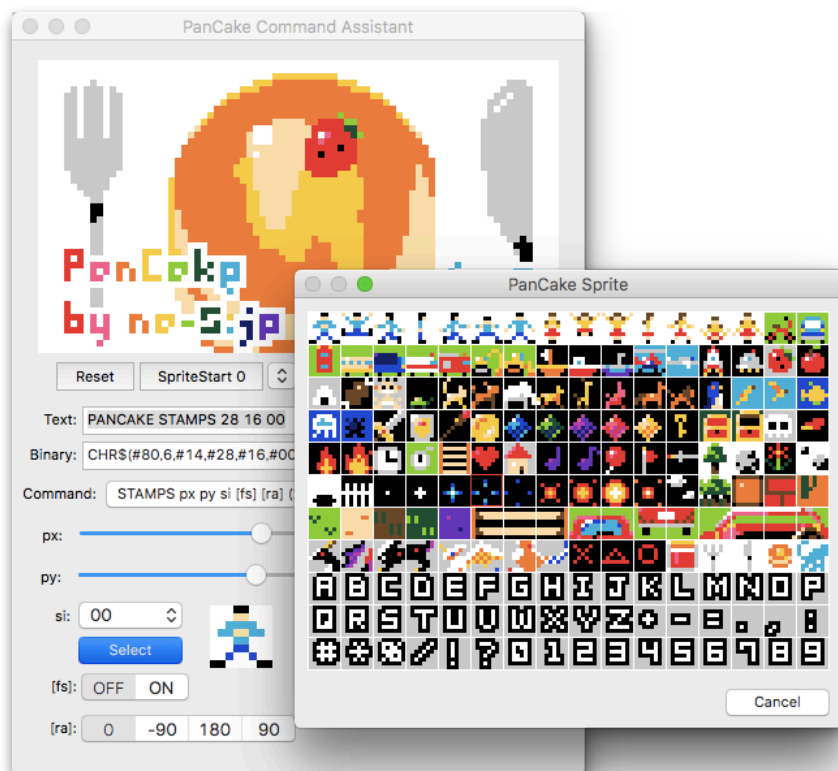
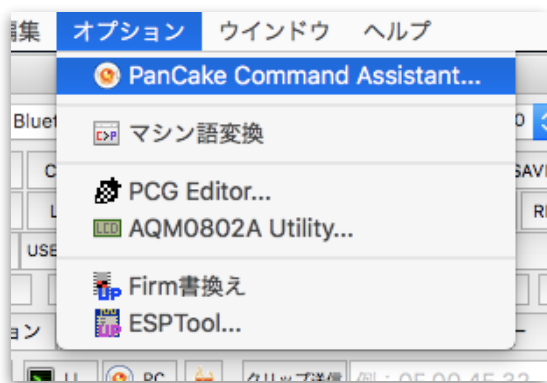
削除

リストから削除したいファイルを選択して下さい。図のようなダイアログがでますので、OKで削除されます。



【PanCake command Assistant】

PanCakeのコマンドを簡単に作る機能を設けました。オプションメニューの「PanCake Command Assistant」を選択するとアシスタントのウィンドウが開きます。



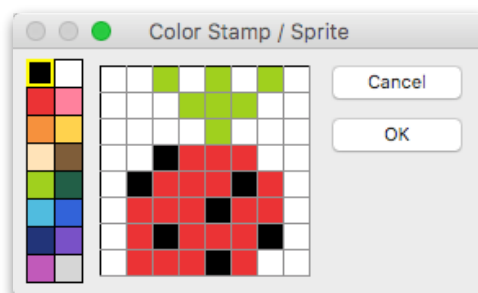
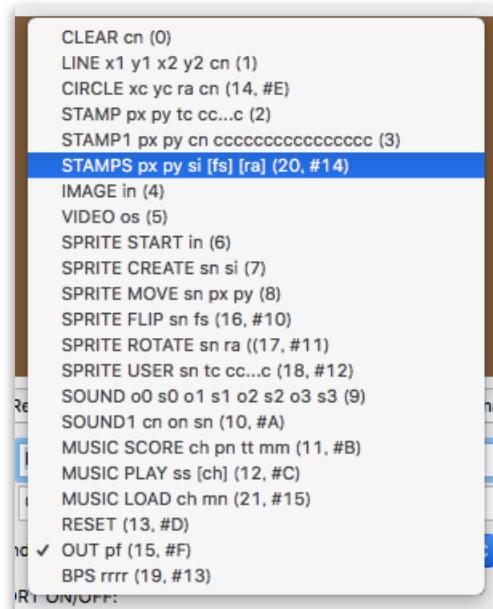
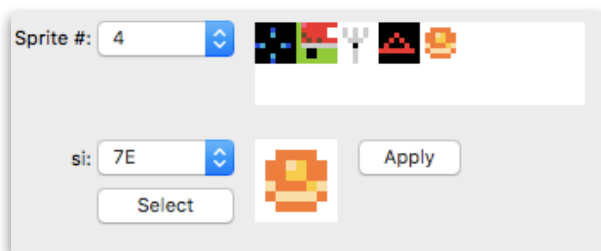
1、「Command」のメニューで作りたいコマンドを選択します。

2、そのコマンドの設定項目を設定すると、テキストとバイナリーのコマンドが作成されます。

3、設定したデータはApplyボタンをクリックしないと、画像に反映されません。

<Sprite Createの設定方法例>

- 1、Sprite #を選択
- 2、Selectボタンを押してスプライトを選択
- 3、「Apply」ボタンをクリック



【マシン語変換（Bin2Pokeの設定及び使い方）】

IchigoJam-FANで上がった「Cで書いたusr()で呼び出すマシン語部を、BASICのpokeに変換するツール」(<https://www.facebook.com/groups/ichigojam/permalink/633543440118746/>)を組み込みました。なお、ハンドアッセンブルする場合には、ARMのツールや、Pythonのインストールは必要ありません。

1、ダウンロード及びインストール

Macでの手順

● bin2poke.py

https://github.com/rohiniku/IchigoJam_bin2poke の

“Download Zip”でダウンロード、解凍したフォルダを適当な場所に置く

＊必要なファイルは、その中のbin2poke.pyのみです。

● ARMのツール

<https://launchpad.net/gcc-arm-embedded/+download> から

gcc-arm-none-eabi-5_4-2016q2-20160622-mac.tar.bz2 (md5)をダウンロード

（これより最新のものがあったらそちらをダウンロード）

解凍して、適当な場所に置く。

＊Macは、Pythonは最初からインストールされてる（かな？）ため、インストールの必要はありません。

Windowsでの手順

● bin2poke.py

https://github.com/rohiniku/IchigoJam_bin2poke の

“Download Zip”でダウンロード、解凍したフォルダを適当な場所に置く

＊必要なファイルは、その中のbin2poke.pyのみです。

● ARMのツールをインストール

<https://launchpad.net/gcc-arm-embedded/+download> から

gcc-arm-none-eabi-5_4-2016q2-20160622-win32.exe (md5)をダウンロード

してインストールする（これより最新のものがあったらそちらをインストール）。

● Pythonをインストール

<https://www.python.org/downloads/windows/> から

Python 2.7.10 - 2015-05-23 の Windows x86 MSI installerまたは

Windows x86-64 MSI installerをダウンロードしてインストールする。

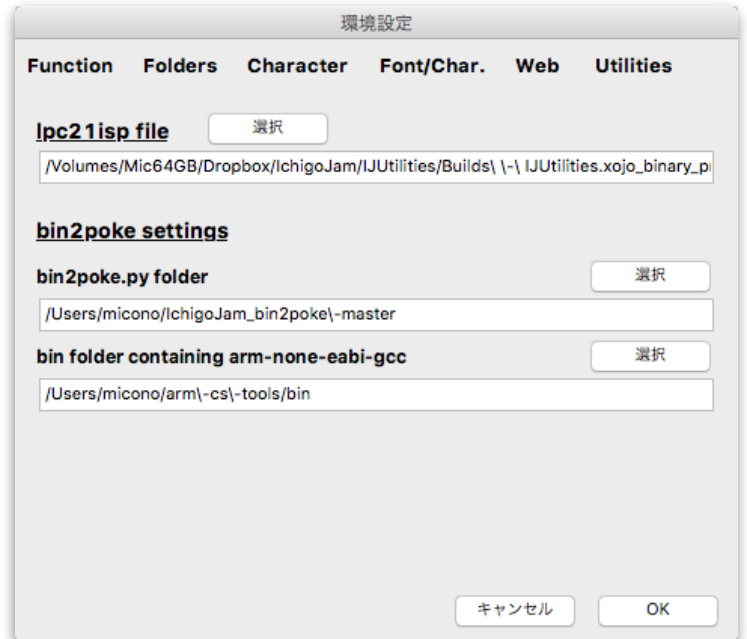
＊3.5.xではうまく動かないみたいです。

＊Cygwinのインストールは必要ありません。

2、設定

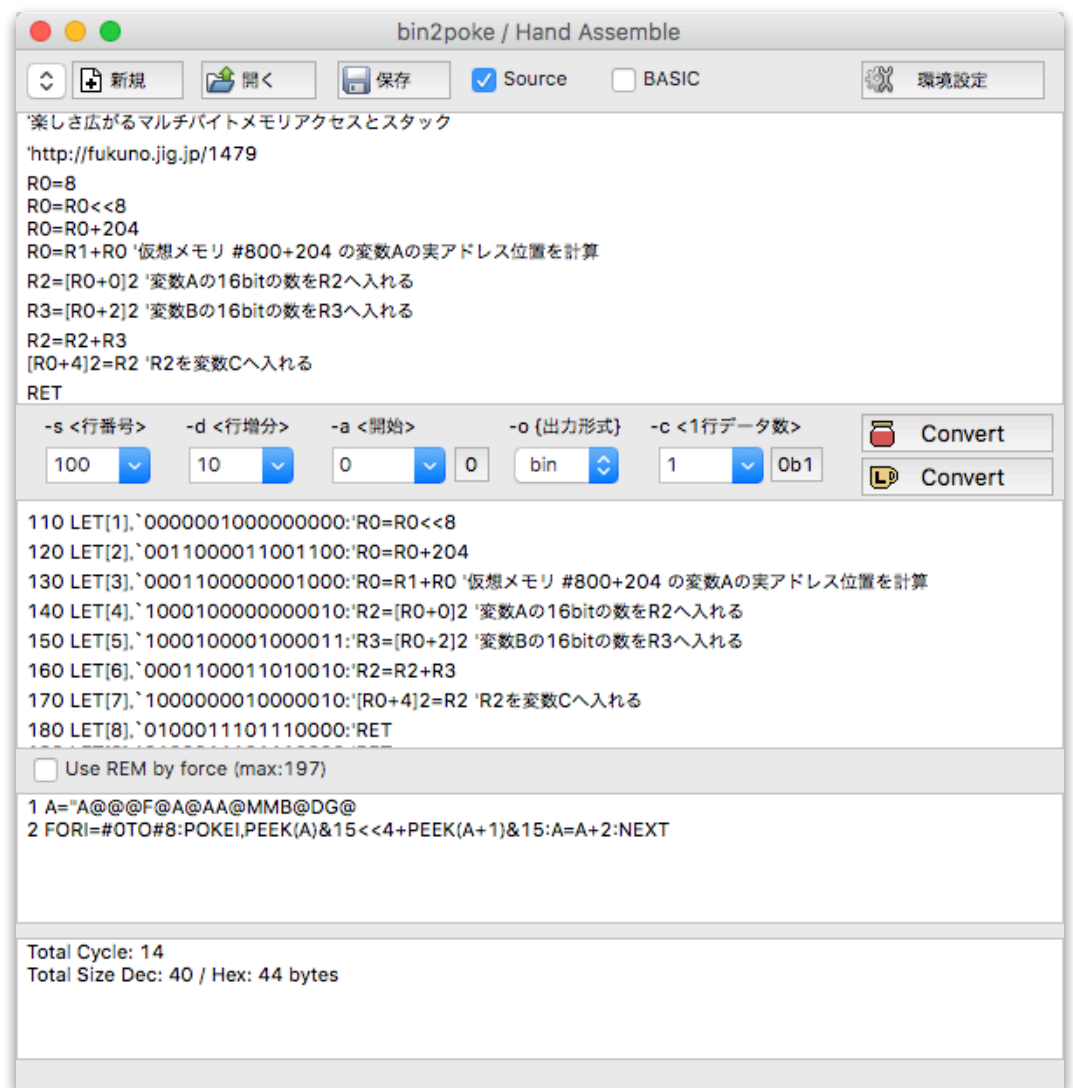
あらかじめ、使用するファイルの入ってるフォルダを設定します。環境設定のUtilitiesでフォルダを設定してください。

- bin2poke.pyの入ったフォルダを選択してください。
- arm-none-eabi-gccとarm-none-eabi-objcopyのファイルを使います。これらの入ったbinフォルダを選択して下さい。Pathが通っていれば指定する必要はありません。
- (Windowsの場合) Python.exeの入ったフォルダを選択して下さい。Pathが通っていれば指定する必要はありません。



3、変換

マシン語に変換したい、Cソースを上のフィールドに書き、IchigoJam用またはIchigoLatte用のConvertボタンのどちらかをクリックすると、マシン語に変換されたソースが下のフィールドに表示されます。なお開始行番号などオプションの設定も気軽に行えます。



4、ハンドアセンブルのマシン語変換について

Bin2Pokeと同じようにハンドアセンブルのマシン語変換ができます。最上部のエディットフィールドにコードを書いて「Convert」ボタンをクリックしてください。

ハンドアセンブルに関しては以下のページを参考にしてください。

- 連載、IchigoJamではじめる、ARMマシン語入門

[1. はじめてのマシン語](#)

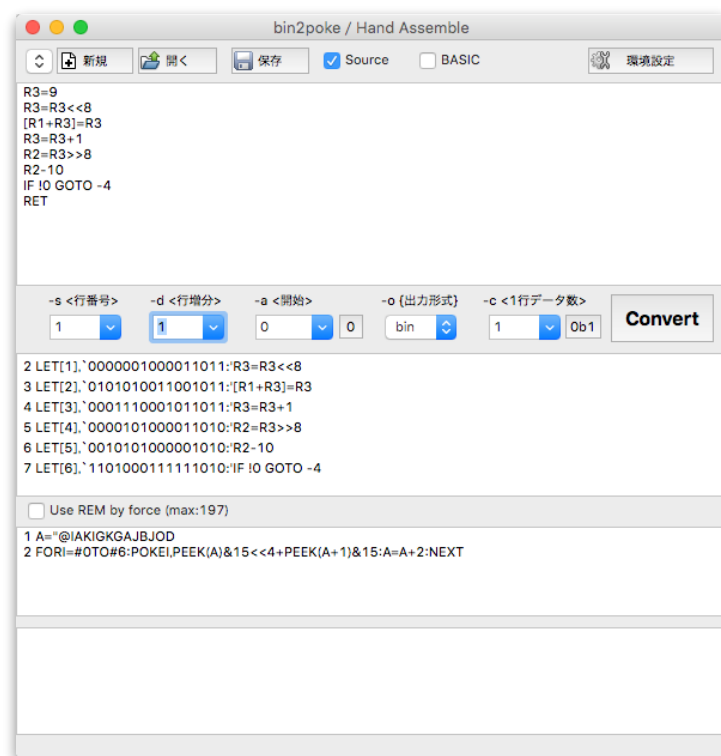
[2. ハンドアセンブルで超速計算！](#)

[3. マシン語メモリアクセスで画面超速表示！](#)

[4. マシン語でLEDを光らせよう！](#)

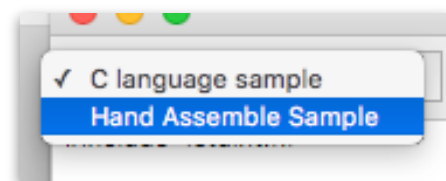
[5. 楽しさ広がるマルチバイトメモリアクセスとスタック](#)

[6. マシン語使いこなしTIPS](#)

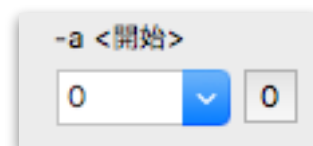


5、その他

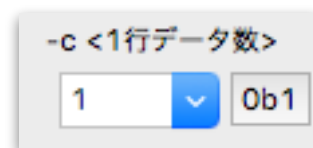
●左上部のポップアップをクリックすると、Cとハンドアセンブルのサンプルソースが読み込めます。



●開始を0にすると、配列に直接マシン語を書き込むタイプのコードに変換されます。



●ハンドアセンブルの場合は、開始が0で1行のデータ数が1の場合、Bin2Pokeの場合は、さらに出力形式がBinの場合には、アセンブリがコメントとして追記されます。



【PCGエディタ】

オプションメニューの「PCG Editor」でキャラクタを編集する為のエディタウィンドウが開きます。8つのキャラクタを同時に編集させられます。選択されているエリアは青色の枠が付いています。編集したいエリアをクリックして青い枠で囲まれた状態にして編集を行ってください。

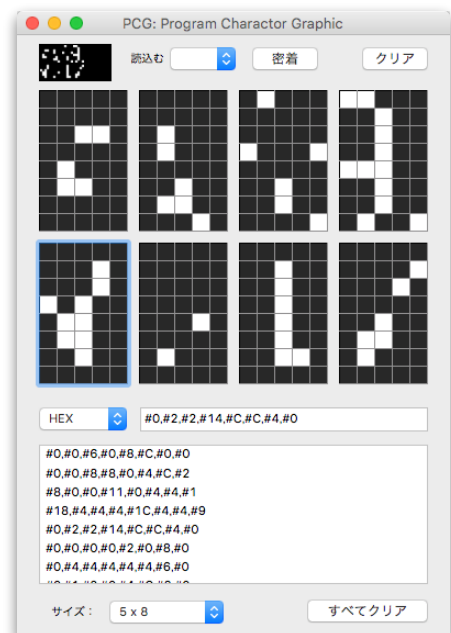
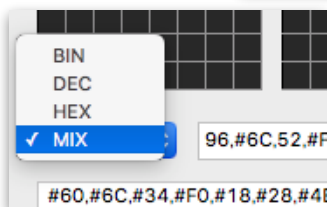
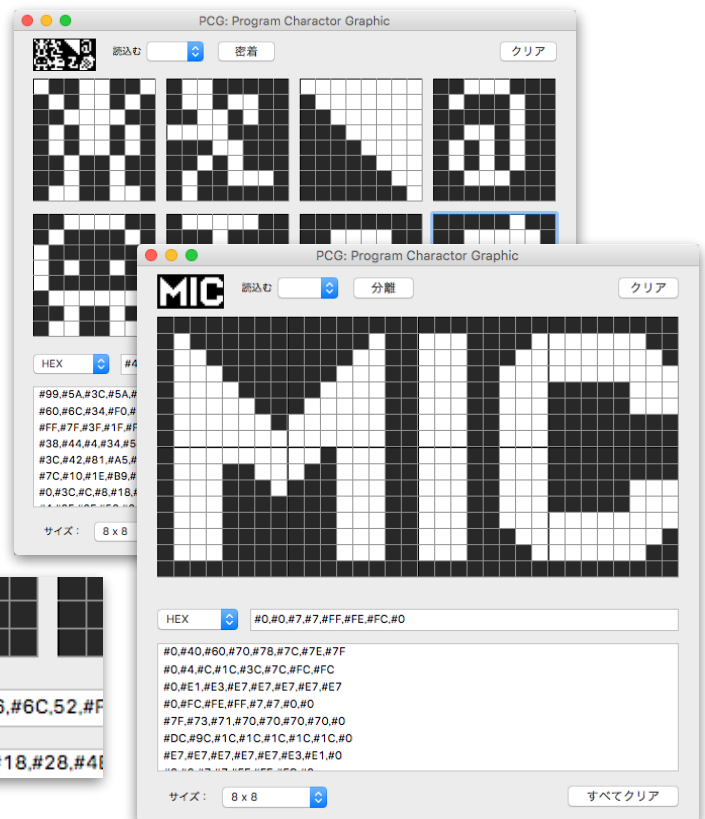
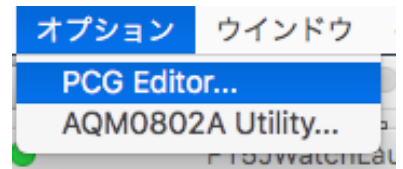
読み込みのポップアップでキャラクタを選ぶとそのキャラクタがコピーされますので、それを参考に編集することもできます。

密着（分離）のボタンで、8つのエリアをくっ付けたり離したりできます。密着にするとエリアどうしのスペースがなくなりますので、必要に応じてくっ付けて編集してください。

コピー、カット、ペーストは、一般的な操作方法でできます。選択されたエリアをコピーして、他のエリアを選択して、ペーストといった使い方をして下さい。また、プログラムで使う形の、
POKE #700,#60,#6C,#34,#F0,#18,#28,#4E,#40
もしくは、キャラクタデータ部分だけの、
#60,#6C,#34,#F0,#18,#28,#4E,#40
といった文字列もペーストできます。

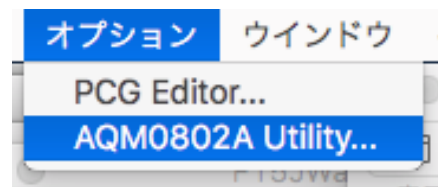
編集結果のデータ形式はBIN,DEC,HEXそしてMIXから選べます。Mixではできるだけ文字数が少なくなるように出力します。

サイズのポップアップで、キャラクタのサイズを8x8と5x8の2種類から選べます。



【AQM0802Aユーティリティ】

オプションメニューの「AQM0802 Utility」でAQM0802A用のコマンドを編集に役立つウィンドウが開きます。



最上段の8このポップアップメニューで、初期化の機能を順に選択することができます。あらかじめ、各種設定を目的の条件に合わせてポップアップで選んで設定して下さい。

各設定の左から並ぶ3つのフィールドはBIN,HEX,DECの順で3種類の値で表示されています。

A screenshot of the 'AQM0802A Utilities' window. At the top, there is a row of 8 dropdown menus: '拡張機...', '内部OS...', 'コント...', 'パワー/!', 'Follow...', '通常機...', 'ディス...', and '画面消去'. Below this is a 'MIX' dropdown and a text field containing 'POKE #700,64,0,2,#C0,57,17,#70,86,#6C,56,#C,1', with an 'Update' button to its right. The main area contains several sections of settings, each with a label, a hex value field, a register number field, and a decimal value field. The sections are: '画面消去' (with a 'ホームに戻る' button), 'エントリーモード設定', 'ディスプレイON/OFF', 'DDRAMアドレスセット', 'ファンクション (機能) 設定', 'カーソル・ディスプレイのシフト (ノーマルモード)', 'CGRAMセット(ノーマルモード)', '内部OSC (拡張モード)', 'ICONアドレス設定 (拡張モード)', 'パワー/ICON/コントラスト設定 (拡張モード)', and 'Followerコントロール (拡張モード)'. Each setting has a corresponding dropdown menu for selection. The '画面消去' section has a blue highlight on the first hex value field.

【MixJuiceなどESP-WROOM-02のファームを書き込む】

特にWindowsの場合は、公式の書き込み用のアプリがあるのであまり問題にならないと思いますが、マックの場合は、コマンドラインでやらなくてはいけないので、このツールを使えば、面倒さを多少軽減できると思います。

準備1、esptool.pyまたはesptool-ckのインストール

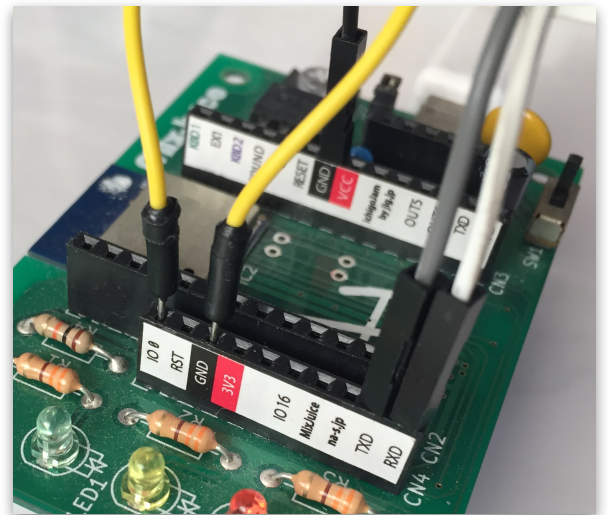
のどちらかを使ってファームを書き込みます。

esptool.pyを使う場合はイチゴジャムレシピさんのページ (<https://15jamrecipe.jimdo.com/mixjuice/>ファームウェアの更新/) を参考にインストールしてください。

esptool-ckを使う場合は、Mac, Win, Linux64版には、IJUtilitiesと同じフォルダにあるesptoolという名前のフォルダにすでにいれてあります。もし最新のものをを使う場合は、igrr/esptool-ck (<https://github.com/igrr/esptool-ck/releases>)からダウンロードしてください。

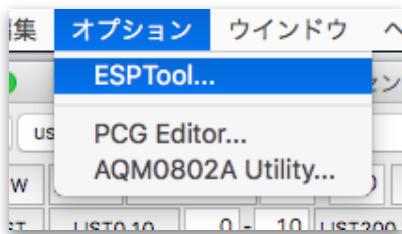
準備2、USBシリアル変換器とMixJuiceを接続する。

- 1、MixJuiceのスイッチをOFFにしてIchigoJamからMixJuiceを外す。
- 2、書き換えのための配線する
MixJuiceのCN4のIO0とどこかのGND
MixJuiceのCN4のRXとUSBシリアル変換器のTX
MixJuiceのCN4のTXとUSBシリアル変換器のRX
- 3、CN4のジャンパをショートして、CN1に電源を繋げる。



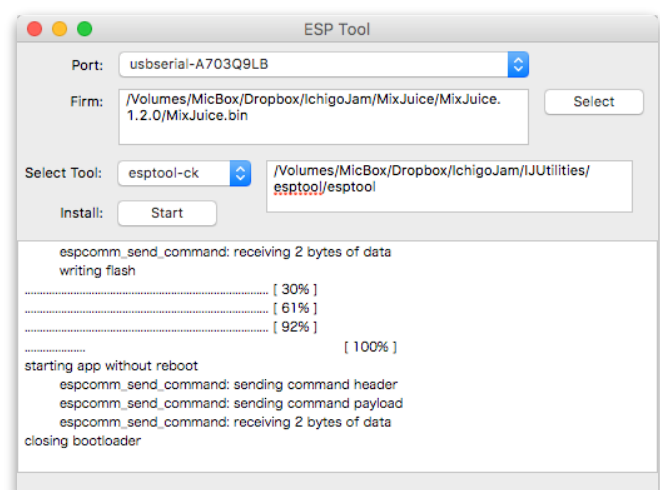
IJUtilitiesでの書き換え操作手順

- 1、ESP Toolを開く



- 2、PortのメニューからUSBシリアル変換器のポートを選択
- 3、SelectボタンでMixJuiceのファームを選択
- 4、ツールを選択*。
- 5、MixJuiceの電源スイッチをオン
- 6、スタートボタンをクリック

*esptool.pyを使う場合は、ターミナルで、`which esptool.py`の検索結果のパスを書き込んでください。



【AQM1248AとIchigoJamをつなぐ】

VDD - VCC

~CS - OUT1（将来は空ける予定）

~RESET - NC

RS - IN4

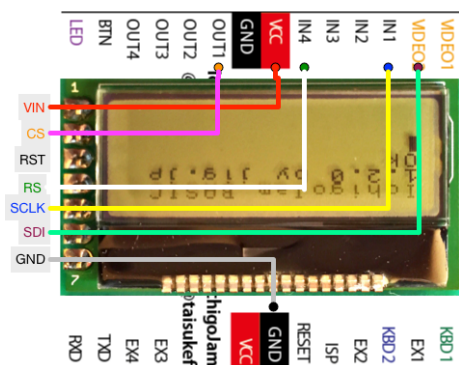
SCLK - IN1

SDI - VIDEO2

GND - GND

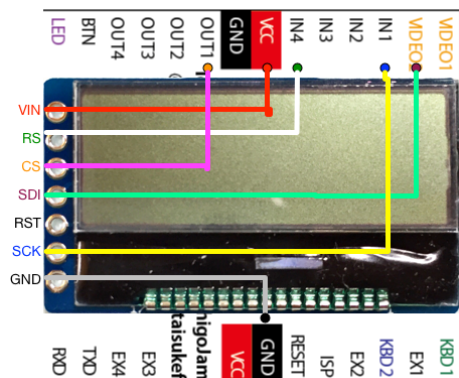
秋月のAQM1248Aの場合（注意：1行目が少し薄いです）

<http://akizukidenshi.com/catalog/g/gK-07007/>



スイッチサイエンスのAQM1248Aの場合（注意：1行目がほとんど見えません）

<https://www.switch-science.com/catalog/2608/>



【IchigonQuest用のBASICについて】

IJUtilitiesでは、IchigonQuestのじゅもんをIchigoJam BASICに準じたIchigonQuest用BASICとして編集します。IchigoJam BASICと同じ様に使えない、IchigonQuest用のBASICとして特別な部分に関して、以下にまとめましたので参考にしてください。

●はこ # について

IJUtilitiesのIchigonQuest用BASICでは、# を[0]（デフォルト）の変数として扱います。

●OUTとOUTPについて

引数がひとつの場合は、OUTに、2つの場合はOUTPに対応します。またOUTPの2つ目の引数のONとOFFはそれぞれ、1と0で示します。

例：

OUT33	5,1をON, それ以外をOFFにする
OUT2,1	2をONにする

●WAITの時間について

WAITには2つの時間単位があります。ひとつはミリ秒、もうひとつはTicks(=1/60秒)です。IJUtilitiesのIchigonQuest用BASICでは、以下のような書式になります。

例：

WAIT30	引数がひとつの場合、IchigoJamと同じ単位はTicksになります。
WAIT500,1000	2つ目の引数を1000とすると単位はミリ秒になります。
WAIT5,60	2つ目の引数を60とすると単位はTicksになります。

●GOIFの書式について

GOIFの書式は以下の様になります。

例：

IF A=2 GOTO 4	はこAが2だった場合、4行目に行きます。
---------------	----------------------

【micro:bit用のプログラムのコマンドリファレンス】

イベント・関数

ずっと	@forever
ボタンAを押した	@ButtonA, @BtnA
ボタンBを押した	@ButtonB, @BtnB
ボタンA+Bを押した	@ButtonAB, @BtnAB
揺らした時	@Shake
ピン0に触れた	@TouchP0, @Tch0
ピン1に触れた	@TouchP1, @Tch1
ピン2に触れた	@TouchP2, @Tch2
ピン0を離した	@ReleasedP0, @Rlsd0
ピン1を離した	@ReleasedP1, @Rlsd1
ピン2を離した	@ReleasedP2, @Rlsd2
ピン0がHighに	@PulseHighP0, @PlsH0
ピン1がHighに	@PulseHighP1, @PlsH1
ピン2がHighに	@PulseHighP2, @PlsH2
ピン0がLowに	@PulseLowP0, @PlsL0
ピン1がLowに	@PulseLowP1, @PlsL1
ピン2がLowに	@PulseLowP2, @PlsL2
無線数値受信	@radioDataReceivedNumber, @radioRcvdNum
無線文字受信	@radioDataReceivedString, @radioRcvdStr
音がなった	@MusicEvent
シリアル受信した	@serialDataReceivedNewLine, @serialRcvdNewLine
シリアル受信した	@serialDataReceivedComma, @serialRcvdComma
シリアル受信した	@serialDataReceivedDollar, @serialRcvdDollar
シリアル受信した	@serialDataReceivedColon, @serialRcvdColon
シリアル受信した	@serialDataReceivedHash, @serialRcvdHash
関数	@関数名

注意： イベントや関数の終わりにはRTNをつけること
関数の呼び出しは、GOSUB @関数名

基本 (basic)

待ち時間	WAIT 30 / pouce 500	basic.pause(500)
LED消去	CLS / clearScreen	basic.clearScreen()
LED表示	showLeds"101011,,,1"	basic.showLeds('# . # . # # ... #')
数字表示	?A / showNumber A	basic.showNumber(A)
文字表示	? "ABC" / showString "ABC"	basic.showString("ABC")
アイコン表示	showIcon "Heart"	basic.showIcon(IconNames.Heart)
矢印表示	showArrow "West"	basic.showArrow(ArrowNames.West)
	showArrow 6	basic.showArrow(ArrowNames.West)

アイコン名称： Heart, SmallHeart, Yes, No, Happy, Sad, Confused, Angry, Asleep, Surprised, Silly, Fabulous, Meh, TShirt, Rollerskate, Duck, House, Tortoise, Butterfly, StickFigure, Ghost, Sword, Giraffe, Skull, Umbrella, Snake, Rabbit, Cow, QuarterNote, EightNote, Pitchfork, Target, Triangle, LeftTriangle, Chessboard, Diamond, SmallDiamond, Square, SmallSquare, Scissors

矢印名称： North, NorthEast, East, SouthEast, South, SouthWest, West, NorthWest

入力 (input)

加速度	A=accel_X	A=input.acceleration(Dimension.X)
-----	-----------	-----------------------------------

	B=accel_Y	B=input.acceleration(Dimension.Y)
	C=accel_Z	C=input.acceleration(Dimension.Z)
	D=accel_Strength	D=input.acceleration(Dimension.Strength)
磁力	A=mag_X	A=input.magneticForce(Dimension.X)
	B=mag_Y	B=input.magneticForce(Dimension.Y)
	C=mag_Z	C=input.magneticForce(Dimension.Z)
	D=mag_Strength	D=input.magneticForce(Dimension.Strength)
傾斜	A=rot_Pitch	A=input.rotation(Rotation.Pitch)
	B=rot_Roll	B=input.rotation(Rotation.Roll)
明るさ	A=lightLevel	A=input.lightLevel()
温度	B=temperature	B=input.temperature()
稼働時間	C=runningTime	C=input.runningTime()
稼働時間	D=runningTimeMicros	D=input.runningTimeMicros()
方向	E=compassHeading	E=input.compassHeading()
方向調整	calibrateCompass	input.calibrateCompass()
ボタン判定	buttonAIsPressed / IsBtnA	input.buttonIsPressed(Button.A)
	buttonBIsPressed / IsBtnB	input.buttonIsPressed(Button.B)
	buttonABIsPressed / IsBtnAB	input.buttonIsPressed(Button.AB)

音楽 (music)

休符	musicRest "Half" / Rest "Half"	music.rest(music.beat(BeatFraction.Half))
	musicRest 2	music.rest(music.beat(BeatFraction.Half))
音を鳴らす	musicRingTone "C"	
	RingTone "C"	music.ringTone(262)
音を鳴らす	musicPlayTone "D","Whole"	
	PlayTone "D","Whole"	music.playTone(
メロディ	musicBeginMelody "PowrUp","Once"	
	BeginMelody "PowrUp","Once"	music.beginMelody(music.builtInMelody(Melodies.PowerUp), MelodyOptions.Once)
テンポ	TEMPO 120	
	musicSetTempo 120	
	SetTempo 120	music.setTempo(120)
テンポ変更	musicChangeTempoBy 20	
	ChangeTempoBy 20	music.changeTempoBy(20)
音符：	1分=Whole, 1/2=Half, 1/4=Quarter, 1/8=Eighth 1/16=Sixteenth, 2分=Double, 4分=Breve	
ドレミ：	<C, <C#, <D, ... C, C#, ... >C, >C#, ... >B	
メロディ：	Dadadadum, Entertainer, Prelude, Ode,Nyan, Ringtone, Funk, Blues, Birthday, Wedding, Funeral, Punchline, Baddy, Chase, BaDing, Wawawawaa, JumpUp, JumpDown, PowerUp, PowerDown	
オプション：	Once, Forever, OnceInBackground, ForeverInBackground	

LED (led)

点灯	ledPlot 2,3 / Plot 2,3	led.plot(2,3)
点灯	ledPlotBrightness 0,0,128	
	PlotBrightness 0,0,128	led.plotBrightness(0,0,128)
点灯が取得	A=ledPoint(2,2)	
	A=Point(2,2)	A=led.point(2,2)
明るさ取得	A=ledBrightness	
	A=Brightness	A=led.Brightness()
消去	ledUnplot 3,1 / Unplot 3,1	led.unplot(3,1)
反転	ledToggle 4,0 / Toggle 4,0	led.toggle(4,0)
動き停止	ledStopAnimation	
	StopAnimation	led.stopAnimation()

表示オン	ledEnable	led.enable(true)
表示オフ	ledDisable	led.enable(false)

無線 (radio)

グループ設定	radioSetGroup 3	radio.setGroup(3)
数値送信	radioSendNumber 3	radio.sendNumber(3)
数値送信	radioSendValue "Hoge",20	radio.sendValue("Hoge",20)
文字送信	radioSendString "MuMu"	radio.sendString("MuMu")
強度設定	radioSetTransmitPower 7	radio.setTransmitPower(7)
シリアル設定	radioSetTransmitSerialNumberTrue	radio.setTransmitSerialNumber(true)
	radioSetTransmitSerialNumberFalse	radio.setTransmitSerialNumber(false)
シリアルに送信	radioWriteReceivedPacketToSerial	radio.writeReceivedPacketToSerial()

ループ

FOR i=0 TO4 … NEXT	For(i=0; i<=4; i++) { … }
--------------------	---------------------------

論理

IF A>0	if(A>0) {
ELIF B<=9	} else if(B<=9) {
ELSE	} else {

入出力端子 (pins)

デジタル	A=IN(0)	A=pins.digitalReadPin(DigitalPin.P0)
デジタル	OUT 0,1	pins.digitalWritePin(DigitalPin.P0, 1)
アナログ	B=ANA(0)	B=pins.analogReadPin(AnalogPin.P0)
サーボ	PWM 0,120	pins.servoWritePin(AnalogPin.P0, 120)
マップ	pinsMap 1,0,1023,20,320	
	Map 1,0,1023,20,320	pins.map(1,0,1023,20,320)
プルアップ	pinSetPullUp 0 / SetPullUp 0	pins.setPull(DigitalPin.P0, PinPullMode.PullUp)
プルダウン	pinSetPullDown 0 / SetPullDown 0	pins.setPull(DigitalPin.P0, PinPullMode.PullDown)
プルアップなし	pinSetPullNone 0 / SetPullNone 0	pins.setPull(DigitalPin.P0, PinPullMode.PullNone)
音を鳴らす	pinsAnalogPitch 0,0 / analogPitch 0,0	pins.analogPitch(0, 0)

シリアル通信 (serial)

通信設定	serialRedirect 0,1,115200	serial.redirect(SerialPin.P0, SerialPin.P1, 115200)
読み込み		
1 行	A=serialReadLine	A=serial.readLine()
改行まで	B=serialReadUntilNewLine	B=serial.readUntil(serial.delimiters(Delimiters.NewLine))
カンマまで	C=serialReadUntilComma	C=serial.readUntil(serial.delimiters(Delimiters.Comma))
\$ まで	D=serialReadUntilDollar	D=serial.readUntil(serial.delimiters(Delimiters.Dollar))
: まで	E=serialReadUntilColon	E=serial.readUntil(serial.delimiters(Delimiters.Colon))
# まで	F=serialReadUntilHash	F=serial.readUntil(serial.delimiters(Delimiters.Hash))
読み込み	G=serialReadString	G=serial.readString()
バッファ	H=serialReadBuffer(64)	H=serial.readBuffer(64)
書き込み		
1 行	serialWriteLine "abc"	serial.writeLine("abc")
数値	serialWriteNumber 2	serial.writeNumber(2)
数値	serialWriteValue "Hoge",20	serial.writeValue("Hoge",20)
文字	serialWriteString "abc"	serial.writeString("abc")
バッファ	serialWriteBuffer(A)	serial.writeBuffer(A)

ゲーム

ゲームオーバー	game.gameOver / gameOver	game.gameOver()
スコア追加	game.addScore 1 / addScore 1	game.addScore(1)
スコア設定	game.setScore 0 / setScore 0	game.setScore(0)
カウントダウン	game.startCountdown 1000 / startCountdown 1000	game.startCountdown(10000)
ゲーム一時停止	gamePause	game.pause()
ゲーム再開	gameResume	game.resume()

数学

絶対値	A=ABS(B)	A=Math.abs(B)
ランダム	B=RND(6)	B=Math.random(6)

＊配列およびBoolean変数は未対応

＊引数を持たない関数にはカッコをつけない

＊引数を持つ関数にはカッコをつける

＊コマンドにはカッコをつけない

＊ボタン判定はIF分の判定で使用可能

A=lightLevel B=mag_Y

A=IN(0) B=ledPoint(2,2)

OUT 0,1 serialRedirect 0,1,115200

if buttonAlsPressed then ...

micro:bit

<http://amzn.to/2DmYPTV>

参考

1. micro:bitではじめるプログラミング —親子で学べるプログラミングとエレクトロニクス

<http://amzn.to/2mV6K3y>

2. micro:bitであそぼう ♪ 導入編: 超かんたん！IoT時代の子ども向けプログラミング学習キット

<http://amzn.to/2DuVSnw>

3. micro:bitであそぼう ♪ 通信編: 超かんたん！IoT時代の子ども向けプログラミング学習キット

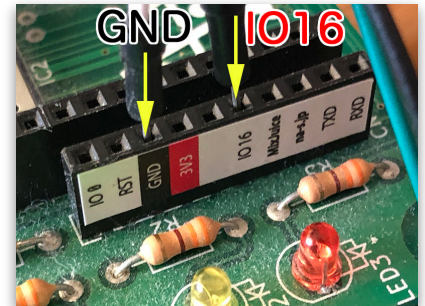
<http://amzn.to/2Dzrjgy>

【MixJuice/MicJackのSoftAPに接続しUDPでデータを送信する】

シリアルでIchigoJamと接続して、ターミナルウィンドウから文字などを入力したり、作ったプログラムを転送したりを、MixJuice/MicJackを介してIchigoJamに無線で送ることができる機能です。

1、MixJuice / MicJackをIchigoJamに繋げて起動

MicJackの場合は最初からSoftAPが動いていますので、IchigoJamと繋いでIchigoJamを起動して下さい。**MixJuice**は起動前にまず、CN4のIO16とGNDをワイヤジャンパーで繋いだ後に起動するとSoftAPが立ちがかります（写真）。



2、MixJuice / MicJackのSoftAPに接続

「MJ-0000000000000000」という名前がMixJuiceまたはMicJackのアクセスポイントになります。0000000000000000の部分はMACアドレスで、**MicJack**の場合は起動時に表示されます。**MixJuice**の場合は、?"MJ MAC"コマンドで確認できます。

3、UDP接続

ツールウィンドウの「UDP」ボタンで接続・切断ができます。



免責など

マニュアル（これ）も未完成です。

クレジット等もまだちゃんとしてません。

アプリケーションの利用および情報のご利用は、自己責任の上でお使いください。

IchigoJam は 株式会社jig.jp の登録商標です。

このアプリは公式のものではありません。このアプリに関するお問い合わせはお問い合わせ先まで。

jig.jpにご迷惑をかけないようにご注意ください。

CC BY: IchigoJam

- ・ IchigoJamから読み込んだキャラクタデータを利用しています。
- ・ IchigoJamの文字をTrueTypeフォントにしたIchigoJam-for-Displayフォントを使用しています。

CC BY / NC: PanCake

- ・ PanCakeのリソース画像を利用しています。

主な公開日

v0.1.0d: 2015/8/16 公開
v0.1.5d: 2015/8/20 公開
v0.2.0d: 2015/8/23 公開
v0.3.0d: 2015/8/30 公開
v0.4.0d: 2015/9/6 公開
v0.4.5d: 2015/9/14 公開
v0.5.0d: 2015/10/18 公開
v0.5.5d: 2015/11/2 公開
v0.6.0d: 2015/12/11 公開
v0.7.0d: 2015/12/31 公開
v0.7.1d: 2016/1/1 公開
v0.8.0d: 2017/2/19 公開
v0.9.0d: 2017/5/3 公開
v0.9.9d6: 2018/9/23 公開

お問い合わせ

micono@mac.com

<http://jjutilities.micutil.com>

過去の更新内容

v0.1.5d

アイコンを変更（まだ未確定）

ターミナルでリターンしても送信されない不具合を修正

Funcボタンで送信できないものがあつたのを修正

Load/Saveなどのファイル番号の修正

LRUN,ECS,HOME,ENDなどFuncボタン追加

カナの送信の不具合を修正（まだ不具合があるかも、カナ以外はまだ未対応）

Windowメニュー、Helpメニュー追加

Import機能追加

正式ファームである1.0.1のピンアサインを仮に作成。1.0.1が正式版とはいえ、公式なものがないので正しいかどうか教えてください。

v0.2.0d

ツールウィンドウ

- ・追加：Importボタンで、ファイルに保存、画像として保存、エディターに表示の2種類でインポートできるようにした。*なお、エディターは、表示できない文字コードを使っている場合には対応していません。
- ・新規：Preferencesで、IchiogaJamの文字を読み込めるようにした。
- ・修正：Preferencesで、BASICコマンドやPINアサインのフォルダを指定していないと落ちてしまう不具合を修正した。

ターミナルウィンドウ

- ・変更：カナ入力ウィンドウで文字を入力すればカナが送信されるようにした。
- ・新規：キャラ入力ボタンをクリックするとキャラクター入力用のウィンドウが表示されます。
- ・変更・修正：通信フィールドで、カナは全角を表示するようにした。表示できない文字はコード番号で表示（送信のみ）。
- ・仮：UART1,UART2ボタンを設置した（仮です）
- ・未対応：表示できない文字はコードの文字の表示はまだ未対応です。

エディターウィンドウ

- ・変更：ファイルの読み書きでUTF-8から生データ読み書きに変更した。
- ・未対応：カタカナ、表示できない文字はコードの文字を含むプログラムの読み書きにはまだ未対応です。

v0.3.0d

ツールウィンドウ

- ・新規：リストビューア：プログラムファイルを入れたフォルダを選択しておくとう便利です。
- ・新規：リファレンス：HTMLファイルを表示します。サンプルとしてIchigoJam-Referというフォルダを用意してあります。
- ・新規：その他リンク：よく使うサイトを環境設定で登録しておけます。

環境設定

- ・追加：リファレンス用のフォルダ選択と、プログラムデータ用のフォルダ選択ができるようになった。
- ・新規：エディタや、印刷時のフォントサイズを設定せきえるようになった。
- ・新規：その他リンクのリンクサイトを編集できます。

エディターウィンドウ

- ・変更：ASCIIタイプ、UTF8、UTF16タイプおよび、ホームページ用の保存ができるようになった。
- ・追加：キャラ文字、カタカナなど、キーボードから入力しにくい文字の入力に対応。
- ・新規：残りバイト数の表示

その他

- ・新規：表示用フォントのIchigoJam-for-Displayフォントを導入（必須）
- ・新規：印刷機能追加

v0.4.0d

ツールウィンドウ

- ・新規：ROMリスト：Filesで読み取ったデータのビューア。メモ書きもしておけます。
- ・変更：読み込みボタンでファイルに保存する機能は削除。エディタに読み込みのみにした。
- ・改良：リストビューアでフォルダ内にも入れるようにした。

環境設定

- ・追加：ROMリスト用のフォルダ選択ができるようにした。

ターミナルウィンドウ

- ・ 変更：キャラクター文字入力のパネルをエディタと同じに変更した。

エディタウインドウ

- ・ 追加：JPEG, PNG, TIFFでの出力機能を追加。
- ・ 新規：折返しなし表示。
- ・ 新規：白黒反転表示

その他

- ・ 変更：Windowsで、MDI表示をやめた。
- ・ 修正：何箇所か文字化け修正

v0.4.5d

フォント

- ・ 改良：IchigoJam-for-Displayの新しいフォントを同梱しました。これまで#80など空欄として表現できてなかった文字に対応しました。
- ・ 新規：モンゴル語文字用のIchigoJam-for-Display-MNを同梱しました。モンゴル語文字を使う場合はフォントのインストールが必要です。

環境設定

- ・ 追加：モンゴル文字対応によりフォントの選択をできるようにした。

ターミナルウインドウ

- ・ 修正：キャラクター入力ボタンをクリックしても、データが送信されない不具合を修正（v0.4.1d）。
- ・ 新規：RENUMボタン追加

エディタウインドウ

- ・ 新規：RENUMボタンを追加。エディタでのRENUMは編集集中のプログラムに対して行われます。またステップ行番号を指定できるとともに、GOTO, GOSUBの行番号の番号振り直しにも対応しています。
- ・ 新規：列と行を表示するようにした。ただし、折返しを解除した状態でないと、本当に正しい値を示しません。

v0.4.6d

フォント

- ・ **修正**：IchigoJam-for-Display/IchigoJam-for-Display-MNのフォントの数値「3」の形状が本来のIchigoJamのフォントの形と異なっていた為、修正したバージョンを同梱しました。（フォントだけは9/20に公開済み）

ターミナルウインドウ

- ・ **修正**：通信データ受信でフラッシュのタイミングが悪く取りこぼしがあったのを修正した。
- ・ 修正：文字表示部分の行間の修正。
- ・ 修正：ESCなどコントロールコードを送ると文字表示部分が文字化けになるのを回避した。

エディタウインドウ

- ・ 修正：文字表示部分の行間の修正。
- ・ 変更：RENUMで正規表現を使うように変更した（内部的）。

v0.5.0d

ツールウインドウ

- ・ **新規**：環境設定のFont/Char設定において、背景色のデフォルト値、シンタックスカラーの設定を可能にした。（シンタックスカラーはマック版のみ）

ターミナルウインドウ

- ・ **新規**：クリップ送信：コピーしてクリップボードに入っているデータを送信する。
- ・ **新規**：UART3ボタンを追加。CR+LF対応不完全
- ・ **改良**：行間を1.0/1.25/1.5の3段階から選択可能にした。

エディタウインドウ

- ・ **新規**：シンタックスカラー対応（マック版のみ）。

v0.5.5d

ツールウインドウ

- ・ **新規**：IchigoJam BASICファーム書き換え機能を追加（lpc21ispボタン）。一応、lpc21ispというフォルダの中にlpc21isp・lpc21isp.exe入れています。**あらかじめ、環境設定のUtilitiesでlpc21ispのファイルを選択しておく必要があります。**
- ・ **新規**：Bin2Poke機能を追加（Bin2Pokeボタン）。IchigoJam-FANで上がった「Cで書いたusr()で呼び出すマシン語部を、BASICのpokeに変換するツール」（<https://www.facebook.com/groups/ichigojam/permalink/633543440118746/>）をIJUtilitiesから使えるようにしました。***各種ファイルのインストールと、環境設定のUtilitiesでフォルダを選択しておく必要があります。**
- ・ **追加**：リストビューアで".c"ファイルを表示・開けるようにした。

v0.5.6d

ツールウインドウ

- ・ **修正**：Bin2Pokeのc言語のエディタで他からコピー&ペースとした場合などハイパーリンクなどが反映されてしまい、マシン語への変換にエラーが出てしまうため、スタイル化されないように修正した。
- ・ **変更**：リストビューアで".c"ファイルを表示させた場合、システムフォントで表示するようにした。

エディターウインドウ

- ・ **修正**：他からコピー&ペースとした場合などハイパーリンクなどが反映されてしまい、シンタックスカラーなどが正しく表示されない不具合を修正した。

v0.5.7d

環境設定

- ・ **修正**：ROMリストフォルダ、lpc2ispファイル、bin2poke.py、ARMのbinフォルダ、を設定しても消えてしまう場合がある不具合を修正した。
- ・ **変更**：タブの名前や順序と内容を少し変更した。
- ・ **新規**：Otherのタブに、Key Bufferという機能を追加、数値を大きくするとプログラムの転送速度が上がるかもしれませんが、取りこぼしが起きるかもしれません（*動作確認してません：初期値は9バイトです）。

v0.6.0d/0.6.1d

エディターウインドウ

- ・ **新規**：DENUM/RENUM機能追加（詳細は後述）
0.6.1d：ラベルを削除した際の不具合修正
- ・ **修正**：【Windows版】画面のアップデートが周期的に起きてしまう不具合を回避した
- ・ **修正**：【Windows版】ウインドウを閉じると落ちてしまう不具合を修正
- ・ **修正**：JPEG/PNG/TIFF画像で出力した場合に文字化けした画像になる不具合を修正

ターミナルセンター

- ・ **修正**：最初の行の文字が少し欠けてしまう不具合を修正

環境設定

- ・ **修正**：Otherのキャラクタ表示が正しく表示されてない不具合を修正
- ・ **修正**：OtherのKeyBufferの数値が保存されない不具合を修正

全般

- ・ **更新**：開発環境のバージョンアップ

v0.7.0d

モニターウインドウ

- ・ **新規**：モニター画面表示に対応しました（詳細は後述）
＊プレイ中画面：3D迷路（SHIROのモバイル日記より）

エディターウインドウ

- ・ **修正**：【Mac版】キャラクターボタンのヘルプタグが表示されない不具合を修正

ターミナルセンター

- ・ **修正**：【Mac版】キャラクターボタンのヘルプタグが表示されない不具合を修正

0.7.1d

- ・ **新規**：受送信したシリアルデータのASCII・HEX処理をオン・オフできるようにした。
- ・ **修正**：SCROLLコマンドで、下向きのスクロールに不具合を修正
- ・ **修正**：モニターウインドウを開いた状態で、キャラクターファイルを読み込んだ場合、キャラクターがモニターウインドウで文字が反映されない不具合を修正した。

0.7.2d

- ・ **修正**：モニター表示でechoオフに対応してなかった不具合を修正
- ・ **修正**：エディターウインドウでCRやCR/LFの場合の転送の不具合を修正

0.7.3d

エディターウインドウ

- ・ **修正**：【Mac版】引用符が“ ”に自動変換されないように対応
- ・ **修正**：キーボードからのカタカナ入力の不具合を修正。
全角で英数字カタカナ入力しても適切な文字に変換します。
- ・ **修正**：印刷で文字化けする不具合を修正
- ・ **修正**：ファイルに読み書きなどでDがdに変換されてしまう不具合を修正

0.7.4d

- ・ **対応**：文字列で引用符(")なしで閉じられている場合の読み書き転送に対応

ターミナルセンター

- ・ **対応**：プログラムをコピー・カットした際、ASCIIでコピーに対応

- ・ **修正**：プログラムをベースとした際、適正な文字に変換に対応
- ・ **対応**：プログラムをベースとした際、データ送信に対応

エディターウィンドウ

- ・ **対応**：プログラムをコピー・カットした際、ASCIIでコピーに対応
- ・ **修正**：プログラムをベースとした際、適正な文字に変換に対応

0.7.5d

- ・ **修正**：0.7.4dの修正で生じてしまったプログラム転送時の不具合に対応

0.7.6d

- ・ **修正**：[Linux系] 通信速度の切り替えの不具合を修正
- ・ **新規**：[Linux系] 通信のための設定方法の説明書を作成
- ・ **改良**：文字処理を改良（新たに不具合を含んでしまっていないか注意してご利用ください）

0.7.7d / 0.7.8d

- ・ **修正**：引用符(")のカラー表示の不具合を修正（0.7.8d）
- ・ **修正**：小文字xが正しく挙動しない不具合を修正
- ・ **修正**：引用符内、コメント文字も含め大文字に変換される不具合を修正
- ・ **新規**：Uppercase/Lowercase変換機能追加

0.7.9d

- ・ **修正**：カタカナなどの文字のコピーおよびペーストの不具合を修正
- ・ **修正**：ファンクション内のボタンをクリックすると落ちる不具合を修正

0.7.10d

- ・ **対応**：ジャンプ先がラベル対応に合わせDENUM/RENUMをラベルに対応
- ・ **修正**：[WIN] エディタで選択しても選択領域が消えてしまう不具合を修正

0.7.12d / 0.7.11d

- ・ **修正**：[WIN版以外] #20-#7Fの文字ボタンをクリックした場合の不具合を修正
- ・ **修正**：エディタウィンドウでアップデートの重複を軽減
- ・ **修正**：エディタウィンドウのアップデートに関する不具合を修正
- ・ **修正**：[WIN版] エディタで文字の選択が正しくできない不具合を修正（再）
- ・ **注意**：[WIN版] エディタでカラー選択できるようになってますが未完成です

0.7.13d

- ・ **対応**：表示フォントを選択可能にした（1）。環境設定のFont/Charのフォントメニューから選択して下さい。UTF 0x100-0x1FFにIchigoJam用のフォントが作られているものが利用できます。
- ・ **新規**：Windows版の64bit版をテスト公開
- ・ **再度**：Linux版の32bit版をテスト公開

0.7.14d

- ・ **対応**：表示フォントを選択可能にした（2）：モニター表示機能で使う文字データを旧タイプフォントと新タイプフォントで環境設定の文字設定で選択できるようにした。（3）：またファイルから読み込んだ場合のデータを初期値として保存できるようにした。
- ・ **修正**：0-0x1Fまでの置換される文字を置換しないようにした。

0.7.15d

- ・ **変更**：BASICコマンドで扱うファイルの文字コードをイチゴジャムレシピさんの変更に伴い、IUtilitiesでもUTF-8対応にした。

0.7.16d

- ・ **新規**：モニター機能で、IchigoJam 1.2.0-big-beta (<https://www.facebook.com/groups/ichigojam/728680847271671/>) の**16x12のBig表示**と、IchigoJam 1.2.0 (正式版：<http://ichigojam.net/farm.html>) にある**16x6の AQM1248A LCD表示**に対応しました。

0.7.17d,18d

- ・ **改良**：Bin2poke機能で、BASE16形式の出力に対応
v.7.18dで出力形式の改良を行いました

0.7.19d, 20d, 21d, 22d, 23d

- ・ 23d：マシン語変換機能で
 - 「7. カジュアルに使うインラインマシン語」で追加された特殊演算を追加
 - 自動クオート変換の処理されてしまう問題を修正
 - 2バイトでHEX出力した場合に、正しい2バイトになってない不具合を修正
- ・ 22d：マシン語変換機能で
 - RET, CPSID, CPSIE, WFI, NOPが含まれていると落ちる不具合を修正

- ・ 21d：マシン語変換機能で
 - ラベル文に対応：書式は「6. [マシン語使いこなしTIPS](#)」を参考にして下さい。ラベルを使う場合、ジャンプ先の数値はなくても大丈夫です。
 - PUSH,POP文でLR,PCが含まれていた場合にR0がONになってしまう不具合を修正。ジャンプ先がプラスの場合の不具合を修正
- ・ 20d：マシン語変換機能で
 - Cortex-M0 ARMマシン語表（抜粋）に対応
 - Rd=[Rn], [Rn]=Rdタイプに対応
 - 「」などによるコメントを使えるようにした
 - Cycle数、バイト数を換算し表示するようにした
 - GOTO文など多数の不具合を修正
- ・ 新規 (19d)：ハンドアセンブルによるマシン語変換機能を追加：

ツールウィンドウから「マシン語変換」ボタンでハンドアセンブルの編集とマシン語への変換ができます。これまでのBin2Pokeと同じエディット画面で同じように利用できます。

(参考ページ)

 - 連載、IchigoJamではじめる、ARMマシン語入門
 1. [はじめてのマシン語](#)
 2. [ハンドアセンブルで超速計算！](#)
 3. [マシン語メモリアクセスで画面超速表示！](#)
 4. [マシン語でLEDを光らせよう！](#)
 5. [楽しさ広がるマルチバイトメモリアクセスとスタック](#)
 6. [マシン語使いこなしTIPS](#)
 7. [カジュアルに使うインラインマシン語](#)

0.7.24d

- ・ 新規：プログラムエディタでプログラム保存形式と同じフォーマットでの出力に対応。JamToast対応のファイルとしてSDにコピーして使えます。
- ・ 新規：FlashAirにファイルをアップロードするためのツールを同梱
- ・ 新規：モニター機能で、20x15 の SPI mobile big に対応。
- ・ 新規：モニター機能で、20x4, 16x2, 8x2の表示に対応。これらのサイズでは画面の上下左右のスクロールはしません。

0.7.25d

- ・ 改良：同梱のFlashAir_toolの表示でアップロード、ダウンロード、消去方法の説明など表示をわかりやすくした。
- ・ 改良：Memory形式での読み書きに対応しました（全バージョンでは出力のみでした）。.BASという拡張子のファイルはデフォルトではMemory形式として認識します。テキスト形式のファイルとして扱いたい場合は環境設定のOtherの中にあるBASの扱いの設定をオンにしてください。
- ・ 新規：リストビューアでファイル形式を変換して保存できるようにした。
- ・ 新規：FlashAirファイルリストを表示させるボタンをつけた。

0.7.26d

- ・ 新規：PCGエディタを追加。8x8および5x8の編集ができます。
- ・ 新規：AQM0802AのLCDのコマンドなどのユーティリティを追加。
- ・ 改良：エディタのRENUM/DENUM機能で、番号対応、ラベル対応の設定がわかりにくかったためポップアップで「#（番号）」と「@（ラベル）」で選べるようにしました。
- ・ 修正：エディタのカラーシンタックスで、幾つかのコマンドのカラーリングに不具合があった為、修正しました。

0.7.27d

- ・ 新規：モニター表示で、VIDEO 2～8のサイズに対応した。
- ・ 新規：モニター表示で、BTN(***）の2回押しに対応した。RUNボタンの左のチェックボックスにチェックを入れておくと、RUN中において、スペースキーと矢印キーが2回押しになります。
- ・ 改良：Firm書換えで、ウィンドウ上でポートの設定、Firmファイルの選択したのちに、インストールするようなかたちに改良した。
- ・ 変更：エディットウィンドウで「すべて送信」を行なった場合、最初にNEWした後に送信するように変更した。

0.7.28d

- ・ 新規：EspToolを使ってMixJuiceなどESP-WROOM-02のファームの書き込みをする機能を追加。
 - ・ 修正：IchigoJamのバイナリー形式のファイルの読み書きで正しく読み書きできない不具合を修正しました。
- [IchigoJam-ap版の読み書きにご利用ください。](#)**

0.7.29d

- ・ 新規：ターミナルウィンドウの「**App**」ボタンからIchigoJam Ap版を起動できるようにした。IJUtilitiesからIchigoJam Apにプログラムの転送など、これまでUARTで繋いで行っていたのと同じ操作がIchigoJam Ap版に対して行えます（WIN版は起動させられるだけで、IJUtilitiesのターミナルウィンドウを介した利用はできません）。

0.8.2d

- ・ 新規：IchigoJam BASIC ver1.2に対応したフォントの日本語版、モンゴル語版、ベトナム語版を作成し、同梱した。

0.8.1d

- ・新規：PanCakeコマンドアシスタントでPanCake用カースタンプ・スプライト編集機能を追加した。
- ・対応：MixJuiceシミュレータ機能において、MixJuice v1.2.1でサポートされたMJ MAC機能に対応した。
- ・修正：MixJuiceシミュレータ機能において改行の不具合を修正。
- ・改良：環境設定で、シリアル通信でDTR, RTSなどの設定ができるようになった。DTR, RTSがONでないと応答しないモジュールとの通信が可能になった。
- ・修正：PanCakeコマンドアシスタントでいくつかの小さな不具合を修正した。
- ・改良：マニュアルに目次を追加

0.8.0d

- ・新規：PanCakeシミュレーション機能追加
- ・新規：PanCakeコマンドアシスタント機能
- ・新規：MixJuiceシミュレーション機能追加
- ・改良：コマンド一覧フォルダを3つまで選べるようにした。

0.9.0d

- ・新規：IchigoLatteに対応しました。
- ・プログラムの編集、読み書き、プログラムの受送信
- ・IchigoLatte用マシン語変換

0.9.1d (未公開)

- ・修正：IchigoJamモードなのにIchigoLatteファンクション表示になってしまう不具合を修正。

0.9.2d

- 新規：プログラム残量表示の1024/4096の切り替えに対応。
(残量表示のところをクリックするとトグルします)
- その他：ハンドアセンブリ機能はすべての機能に対応していません。

0.9.3d

- 修正：エディタで、プログラムをコピーした場合に、改行が変になってしまう不具合を修正

0.9.4d

- 修正：「読み込み」でプログラムを読み込んだ際の文字コードの不具合を修正
- 新規（マック）：TXT, BINファイルなどファイルをアイコンにドロップして開けるようにした

0.9.5d (Facebookでリンク先公開)

- 修正：メモリー形式の読み書きの不具合を修正

0.9.6d

- 新規：一般的な検索&置換の機能を追加（メニューの編集→検索&置換）
- 新規：使用している変数・配列の一覧表示の機能を追加（Jamのみ、Latte非対応）
- 新規：使用している変数の検索と置換の機能を追加（Jamのみ、Latte非対応）
- 新規：プログラム内容のリスト化する機能を追加
- 修正：メモリー形式の読み書きの不具合を修正（ホームページで公開）

0.9.7d

- 新規：ネットを介した読み書きに対応（FlashAirへの読み書きも含む）
(ダウンロードは現在Http GETによるダウンロードのみ。FTPによるダウンロードも対応予定)
- 詳しくは【ネットを介したファイルの読み書き】をご覧ください。

0.9.8d1 (2018/1/1): IchigoJamのコマンドで使えるようなコマンドに対応

- 新規：IchigoJamのプログラムをmicro:bit Javascriptに変換する機能を追加

0.9.8d3 (2018/1/8): micro:bit固有のコマンドも含め多く対応

0.9.8d4 (2018/1/20): バグ修正

0.9.8d5 (2018/1/21): ボタン判定に対応、バグ修正

0.9.7d

新規：ネットを介した読み書きに対応（FlashAirへの読み書きも含む）

（ダウンロードは現在Http GETによるダウンロードのみ。FTPによるダウンロードも対応予定）

詳しくは【ネットを介したファイルの読み書き】をご覧ください。

0.9.6d

新規：一般的な検索&置換の機能を追加（メニューの編集→検索&置換）

新規：使用している変数・配列の一覧表示の機能を追加（Jamのみ、Latte非対応）

新規：使用している変数の検索と置換の機能を追加（Jamのみ、Latte非対応）

新規：プログラム内容のリスト化する機能を追加

修正：メモリー形式の読み書きの不具合を修正（ホームページで公開）

0.9.8d5 (2018/1/21): ボタン判定に対応、バグ修正

0.9.8d4 (2018/1/20): バグ修正

0.9.8d3 (2018/1/8): micro:bit固有のコマンドも含め多く対応

0.9.8d1 (2018/1/1): IchigoJamのコマンドで使えそうなコマンドに対応

新規：IchigoJamのプログラムをmicro:bit Javascriptに変換する機能を追加

0.9.9d1 (2018/2/11)

修正：マック版でダウンロード・アップロードができない不具合を修正

更新：IchigoJamのプログラムをmicro:bit Javascriptに変換する機能の追加と更新

新規：IchigonQuestに対応（編集、受信、送信、フォント）

0.9.9d2 (2018/4/22)

更新：IchigonQuestのOUT/OUTPに対応

更新：コマンド一覧の設定を3つから6つに増やした

修正：Windows版でIchigonQuestからIJUtilitiesへ転送時の不具合を修正

修正：IchigonQuestのWAITのミリ秒とTicksの扱いのバグを修正

修正：IchigonQuest用のプログラムを開く時にIchigoJamとして開かない仕様にした

0.9.9d3 (2018/6/3)

修正：エディタで変数の一覧が出なくなっていた不具合を修正

修正：Windows, LinuxでFirm書換え機能が正しく動かない場合がある不具合を修正（動作未テスト）

0.9.9d6 (2018/9/23)

新規：IchigonQuest風オリジナルのプログラムに対応

新規：IchigonQuestのHEXプログラム表示に対応

修正：IchigonQuest-for-Displayフォント修正（インストールし直しが必要です）

修正：IchigonQuest BAISの変換の不具合をいくつか修正

改良：ESPToolを改良。

0.9.9d7 (2018/9/24)

修正：フォント名がIchigonQestになっていたので、IchigonQuest-for-Displayに名称を直しました。IchigonQest-for-Displayをインストールしている場合は削除して下さい。

修正：IchigonQuest-for-Displayフォントの名称だけでなく内容も改良しました。IchigonQuest-for-Displayフォントをインストールして下さい。

修正：IchigonQuestの3タイプの保存で、HEX, BASICを選んでも、Journeyタイプで保存されてしまう不具合を修正。

修正：IchigonQuestのどうぐ「はこ」の表示がされない場合がある不具合を修正。

改良：IchigonQuest Journeyタイプの保存で、IchigonQuestフォントでなくても読めるような形で出力するように改良。

新規：IchigonQuest Journeyタイプの内容をコピーできるようにした。