

2024/01/27 @ エンジニア作業飲み集会

ゲームボーイアプリ 開発の始め方

kimkim0106



おしながき

- ゲームボーイとは？
 - ゲームボーイの特徴
 - ハードウェアについて
- ゲームボーイ開発に必要なもの
 - 開発環境
 - エミュレーター
 - Flashカートリッジ
- 実際にやってみた

自己紹介



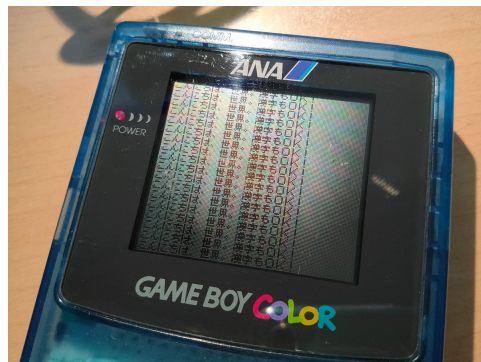
Webエンジニア

kimkim0106

VRChat : kimkim0106
Website : <https://kimkim0106.net>
X(Twitter) : @kimkim0106_3218



こんなことをやっています



ゲームボーイとは？

任天堂の携帯型ゲーム機

- 1989年発売
- 当初はモノクロ画面
- 後にカラーモデルが登場
- 販売台数は1億1869万台
 - ちなみに...
 - DS: 1億5402万台
 - Switch: 1億3246万台

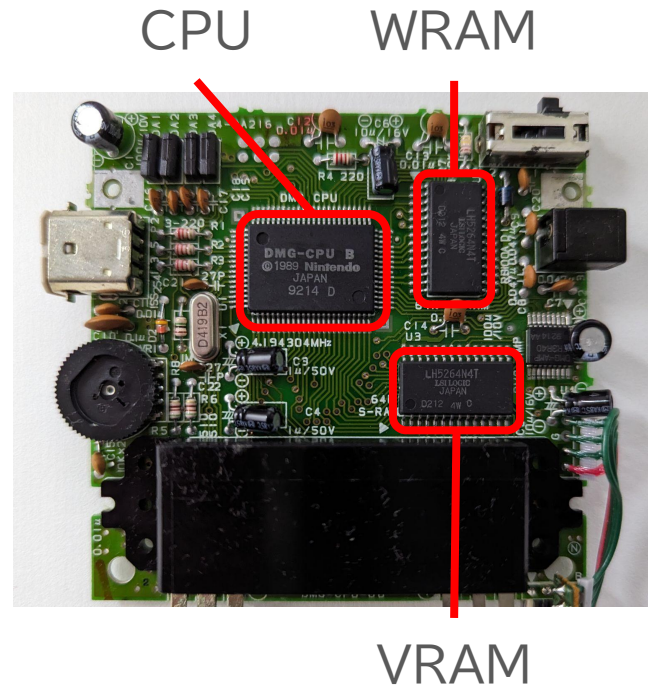


ゲームボーイの特徴

- 長時間遊べる
 - 単3電池4本で35時間、ただし画面はモノクロ
 - 競合機種はカラーだが、単3電池6本で3～4時間
- 乾電池で動く
 - 入手性がよく、交換もしやすい
- 持ち運びしやすい
 - 初代はちょっと重くて大きいけど...
 - ポケット以降は小さくなった

ゲームボーイのハードウェア

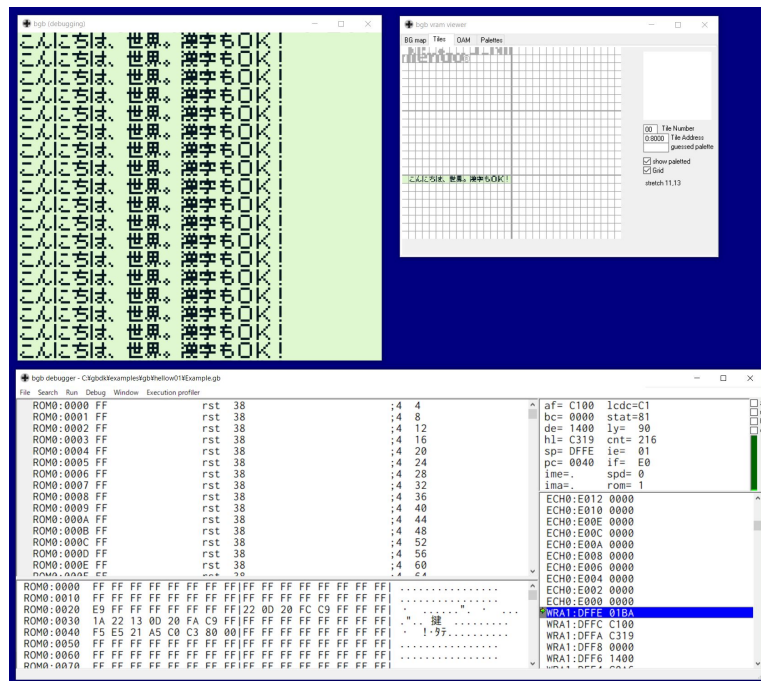
	ゲームボーイ(DMG-01)	ゲームボーイカラー
CPU	SHARP 8-bit CPU (8080 like) @ 4.19 MHz	SHARP 8-bit CPU (8080 like) @ 4.19 MHz, 8.39 MHz
WRAM	8 KiB	32 KiB
VRAM	8 KiB	16 KiB
Screen	モノクロ4階調 液晶 160 x 144	32768色カラー TFT液晶 160 x 144
Sound	4チャンネル ステレオ出力	4チャンネル ステレオ出力
Power	DC 6V 0.7W (単3電池 x 4)	DC 3V 0.6W (単3電池 x 2)



ゲームボーイ開発に必要なもの

非公式だが、一通り揃っている

- 最低限必要なのは...
 - 開発環境
 - エミュレータ
- 実機で動かすなら...
 - Flash カートリッジ
- その他
 - ハードウェア仕様書



開発環境

- Windows・macOS・Linux なんでもOK！
- ツールによってはコーディング不要！
- プログラミングするなら、C言語もしくはアセンブラ言語
- ハードウェアの制約が大きいので、そこは頑張りましょう
 - ROMが小さい
 - グラフィックも自由に書けるわけではない

GB イメージビューワーコンバーター

- 画像を表示させるならコンバーターを使えばOK
- 画像をアップロードするだけ
- ただし4色に減色される

A screenshot of a web browser showing the 'GB イメージビューワーコンバーター ver.0.2' website. The page has a white background with blue accents. It features a list of instructions in Japanese, a file upload section with a preview of a character image, and a '変換' (Convert) button. The footer includes 'CUBIC STYLE 2020'.

GB イメージビューワーコンバーター ver.0.2

- 画像をGBファイルに変換してダウンロードできます。
- サイズは今のところ 160x144 にだけ対応しています。
- カラー画像は 4 階調に変換されます。
- カラーGB表示用のカラーパレットを指定できます。
- 色数が多い画像は「減色時に誤差拡散する」をチェックすると、わりとキレイに表示できます。

ファイルを選択 SBpdEBa...0x144.png

カラーGB用パレット:

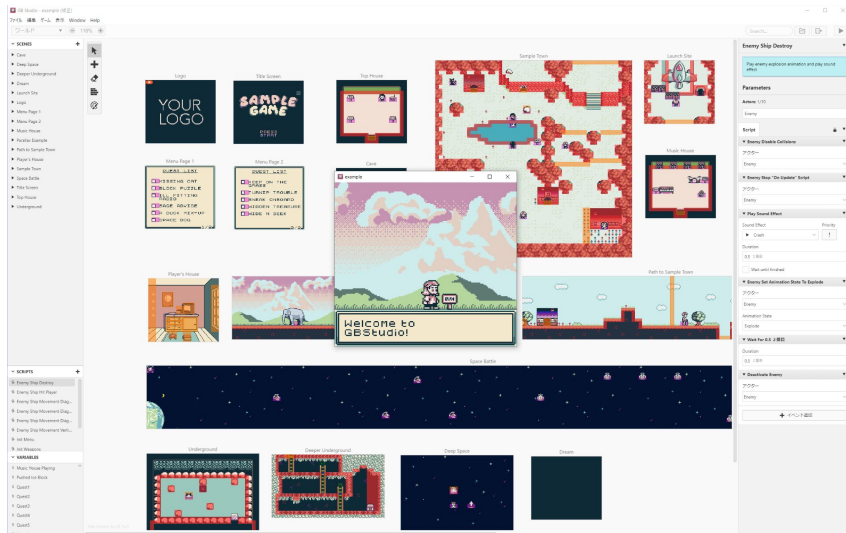
減色時に誤差拡散する
 4階調グレースケールをなるべくキレイに表示する

変換

CUBIC STYLE 2020

GB Studio

- ゲームをドラッグ&ドロップだけで作れる
 - RPGツールの的なソフトだが、RPG以外も作れるっぽい
- Windows・Linux・Mac版あり
- 割と何でもできそう
 - 曲作りもできる



GBDK-2020

- Cプログラミングするならこれ
- 全部入りのクロスコンパイル開発環境
 - ライブラリ・ツールチェーン・コンパイラ
- サンプルも充実
- ZGBを使えばちょっと楽に書けそう
 - 便利な関数が用意されている

GBDK-2020

GBDK is a cross-platform development kit for sm83, z80 and 6502 based gaming consoles. It includes libraries, toolchain utilities and the [SDCC](#) C compiler suite.

Supported Consoles: [\(see docs\)](#)

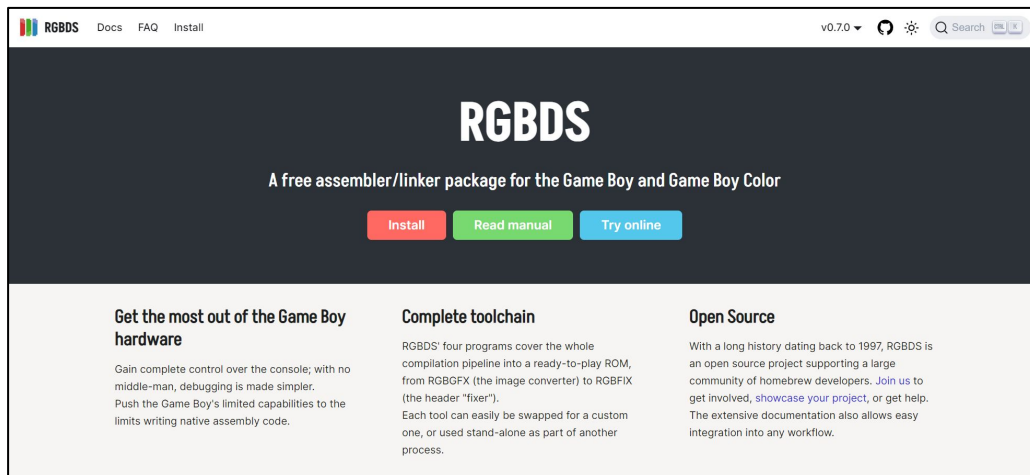
- Nintendo Game Boy / Game Boy Color
- Analogue Pocket
- Sega Master System & Game Gear
- Mega Duck / Cougar Boy
- NES

Experimental consoles (not yet fully functional)

- MSXDOS

RGBDS

- アセンブラ・リンカのパッケージ
- Windows・macOS・Linux対応
 - Arch Linux なら公式リポジトリにある
 - macOSもHomebrewから入れられる



The screenshot shows the RGBDS website homepage. At the top, there is a navigation bar with the RGBDS logo, links for 'Docs', 'FAQ', and 'Install', and a version indicator 'v0.7.0'. The main heading is 'RGBDS' in large white letters, followed by the subtitle 'A free assembler/linker package for the Game Boy and Game Boy Color'. Below this are three buttons: 'Install' (red), 'Read manual' (green), and 'Try online' (blue). The page is divided into three columns of text:

- Get the most out of the Game Boy hardware**
Gain complete control over the console; with no middle-man, debugging is made simpler. Push the Game Boy's limited capabilities to the limits writing native assembly code.
- Complete toolchain**
RGBDS' four programs cover the whole compilation pipeline into a ready-to-play ROM, from RGBGFX (the image converter) to RGBFIX (the header "fixer"). Each tool can easily be swapped for a custom one, or used stand-alone as part of another process.
- Open Source**
With a long history dating back to 1997, RGBDS is an open source project supporting a large community of homebrew developers. [Join us](#) to get involved, [showcase your project](#), or get help. The extensive documentation also allows easy integration into any workflow.

エミュレータ

なんでもいいがデバッグしやすいものがよさそう

- bgb
 - Windows のみだが、Wineでも動く
 - かなり正確
 - デバッグ機能付き
- 他にも色々ある
 - Sameboy (Windows・macOS)
 - VisualBoyAdvance-M (Windows・macOS・Linux)

実機で動かしたい

Flashカートリッジとリーダーライターが必要

- 市販品
 - 同人サークル「CUBIC STYLE」のものがBoothで買える
 - GBカートフラッシャー開発セット
 - 大容量・RAM付きなど色々ある
 - 他にも海外製品が色々ある
- 自作
 - 基板データなど公開されている

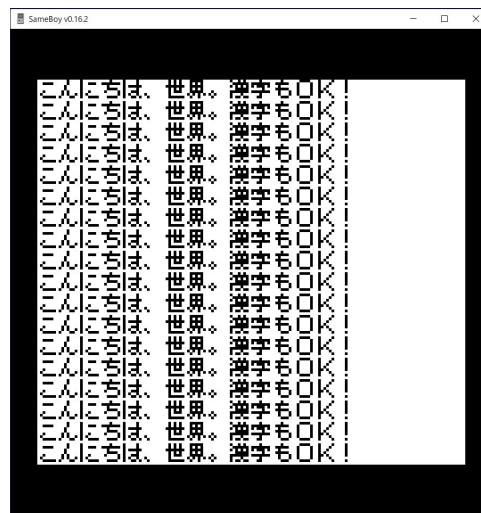


ハードウェア仕様書

- ゲームボーイのハードウェアはかなり解析済み
 - CPUからシリコンを取り出して回路まで解析されようとしている
- 周辺機器も解析されているので使える
 - ポケットカメラ
 - ポケットプリンタ
 - モバイルアダプタGB
- 大抵のドキュメントはインターネット上にある
 - 割りとなんでも出てくる

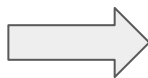
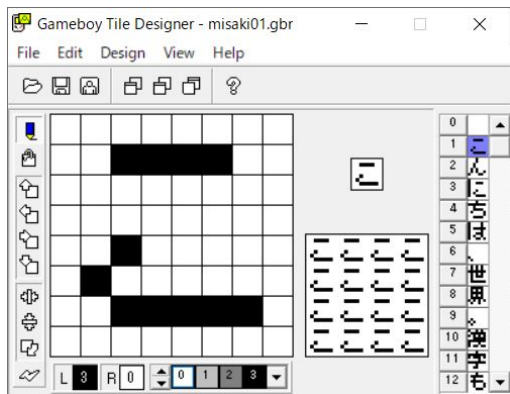
実際にやってみた

- ゲームボーイにグラフィックを表示させるプログラム
 - といいつつ、文章なのですが
 - 英数字であれば、GBDKでprintfするだけで表示できる
- やること
 - データを用意
 - 今回は漢字
 - 並べる
 - 文章になるように並べる
 - ビルドする



① タイルデータを作成

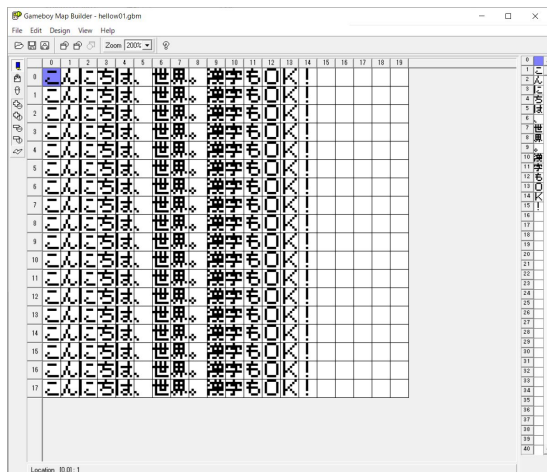
8x8 もしくは 8x16 でタイルデータを作成



```
26 unsigned char Tile[] =
27 {
28     0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
29     0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
30     0x00,0x00,0x3C,0x3C,0x00,0x00,0x00,0x00,
31     0x20,0x20,0x40,0x40,0x3E,0x3E,0x00,0x00,
32     0x10,0x10,0x10,0x10,0x20,0x20,0x20,0x20,
33     0x50,0x50,0x52,0x52,0x8C,0x8C,0x00,0x00,
34     0x80,0x80,0x9C,0x9C,0x80,0x80,0x80,0x80,
35     0x90,0x90,0xA0,0xA0,0x9E,0x9E,0x00,0x00,
36     0x10,0x10,0xFE,0xFE,0x20,0x20,0x3C,0x3C,
37     0x42,0x42,0x02,0x02,0x3C,0x3C,0x00,0x00,
38     0x84,0x84,0x84,0x84,0xBE,0xBE,0x84,0x84,
39     0x9C,0x9C,0xA4,0xA4,0x9A,0x9A,0x00,0x00,
40     0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
41     0x00,0x00,0x80,0x80,0x40,0x40,0x00,0x00,
42     0x54,0x54,0x54,0x54,0xFE,0xFE,0x54,0x54,
43     0x5C,0x5C,0x40,0x40,0x7E,0x7E,0x00,0x00,
44     0x7C,0x7C,0x54,0x54,0x7C,0x7C,0x54,0x54,
45     0x7C,0x7C,0xAA,0xAA,0x48,0x48,0x00,0x00,
46     0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
47     0x40,0x40,0xA0,0xA0,0x40,0x40,0x00,0x00,
48     0x94,0x94,0x7E,0x7E,0xAA,0xAA,0x3E,0x3E,
49     0xFE,0xFE,0x88,0x88,0xB6,0xB6,0x00,0x00,
50     0x10,0x10,0xFE,0xFE,0xBA,0xBA,0x08,0x08,
51     0xFE,0xFE,0x10,0x10,0x30,0x30,0x00,0x00,
52     0x10,0x10,0x7C,0x7C,0x20,0x20,0x7C,0x7C,
53     0x22,0x22,0x22,0x22,0x1C,0x1C,0x00,0x00,
54     0x3C,0x3C,0x42,0x42,0x42,0x42,0x42,0x42,
55     0x42,0x42,0x42,0x42,0x3C,0x3C,0x00,0x00,
56     0x42,0x42,0x44,0x44,0x48,0x48,0x50,0x50,
57     0x68,0x68,0x44,0x44,0x42,0x42,0x00,0x00,
58     0x10,0x10,0x10,0x10,0x10,0x10,0x10,0x10,
59     0x10,0x10,0x00,0x00,0x10,0x10,0x00,0x00
60 };
```

②マップデータ(背景)を作成

タイルをしきつめていく



```
25 unsigned char HELLOW0[] =
26 {
27   0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A,
28   0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x00, 0x00, 0x00, 0x00,
29   0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A,
30   0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x00, 0x00, 0x00, 0x00,
31   0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A,
32   0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x00, 0x00, 0x00, 0x00,
33   0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A,
34   0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x00, 0x00, 0x00, 0x00,
35   0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A,
36   0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x00, 0x00, 0x00, 0x00,
37   0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A,
38   0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x00, 0x00, 0x00, 0x00,
39   0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A,
40   0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x00, 0x00, 0x00, 0x00,
41   0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A,
42   0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x00, 0x00, 0x00, 0x00,
43   0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A,
44   0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x00, 0x00, 0x00, 0x00,
45   0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A,
46   0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x00, 0x00, 0x00, 0x00,
47   0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A,
48   0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x00, 0x00, 0x00, 0x00,
49   0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A,
50   0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x00, 0x00, 0x00, 0x00,
51   0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A,
52   0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x00, 0x00, 0x00, 0x00,
53   0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A,
54   0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x00, 0x00, 0x00, 0x00,
55   0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A,
56   0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x00, 0x00, 0x00, 0x00,
57   0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A,
58   0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x00, 0x00, 0x00, 0x00,
59   0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A,
60   0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x00, 0x00, 0x00, 0x00,
61   0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A,
62   0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x00, 0x00, 0x00, 0x00,
63   };
```

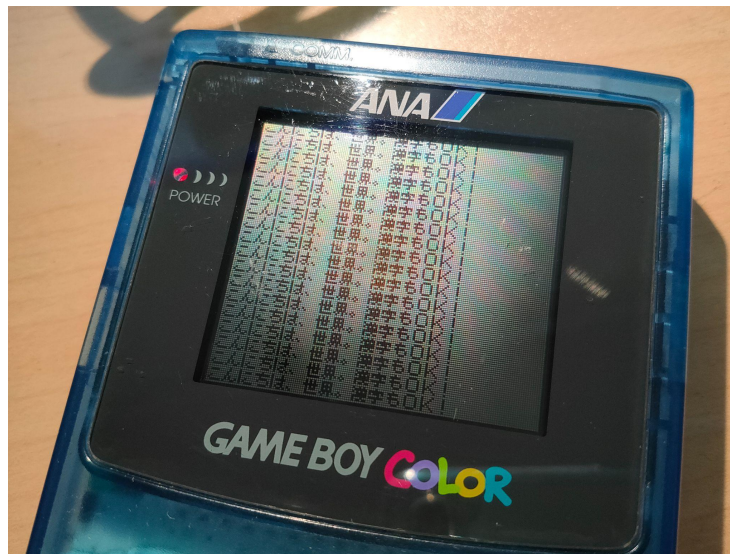
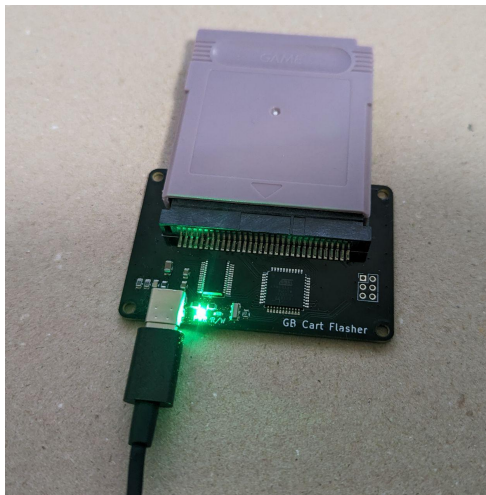
③mainを記述

GBDKに便利な関数があるのでこれだけでいい

```
1  #include <stdio.h>
2  #include <gb/gb.h>
3  #include "tile.c"
4  #include "hellow01.c"
5
6  void main(void) {
7      set_bkg_data(0, 16, Tile);          /* タイルデータを背景用として設定する */
8
9      set_bkg_tiles(0, 0, 20, 18, HELLOW01); /* マップデータを設定する */
10     SHOW_BKG;      /* 背景表示 */
11 }
12
```

④ビルドする

ROMイメージが生成されるので、これを実行する



まとめ

- ゲームボーイはまだまだアツい！
 - 最近、中古価格が高騰しててつらい...
 - 画面の劣化もつらい...
 - 持っている人は大事にしましょう
- みんなもゲームボーイソフト開発をしよう！
 - 自分もがんばります
 - (次のリアルVketまでになにかできるといいな)

リンク

- GB イメージビューワーコンバーター
<https://cubic-style.jp/gbiv/>
- GB Studio
<https://www.gbstudio.dev/>
- GBDK-2020 | gbdk-2020
<https://gbdk-2020.github.io/gbdk-2020/>
- Welcome! | RGBDS
<https://rgbds.gbdev.io/>
- BGB GameBoy Emulator
<https://bgb.bircd.org/>
- SameBoy
<https://sameboy.github.io/>
- CUBIC STYLE - BOOTH
<https://cubic-style.booth.pm/>
- Projects | gbdev.io
<https://gbdev.io/>
- Game Boy: Complete Technical Reference
<https://gekkio.fi/files/gb-docs/gbctr.pdf>

参考文献

- 株主・投資家向け情報:業績・財務情報 - ゲーム専用機販売実績
https://www.nintendo.co.jp/ir/finance/hard_soft/index.html
- ゲームボーイ全機種仕様対比表
<https://www.nintendo.co.jp/n02/dmg/hardware/gbtaihi/index.html>
- ゲームギア | セガ SEGA
<https://www.sega.jp/history/hard/gamegear/index.html>
- Specifications - Pan Docs
<https://gbdev.io/pandocs/Specifications.html>
- Game Boy (DMG) - Game Boy hardware database
<https://gbhwdb.gekkio.fi/consoles/dmg/>

おわり

ご清聴ありがとうございました

資料はWebsiteにて公開します！

<https://kimkim0106.net>