

**Twitter をしよう！**

ぺんた @plageoj

Twitter をしよう！

\$ whoami

ぺんた

@plageoj [Twitter](#) / [GitHub](#) / [Facebook](#) / [Qiita](#)

広島大学工学部 3 年です。  
専門は電子系。趣味は情報系。

一部界限では8bit マシンで [Twitter](#) をやる人として  
知られています。

# TL;DR

8bit マシンで Twitter の検索・投稿ができるようになるまでの努力と涙と徹夜の記録です。

- OSI 参照モデル 5～7 層
- ちょっとした電子工作
- 基本的なコンピュータ構成の知識

Twitter で就活垢を作って云々という話はしません。

Twitter をしよう！

# ことはじめ

かっこええや  
ん、、、

出典 [Tweet](#)

ぺんた (@plageoj) ・ 広島大学工学部



絵を描く PETER

@peter6409

スマホがない世界のSNS端末の絵を描いた。

カセットによって色が変わる

[Translate Tweet](#)



Twitter をしよう！

# ターゲットマシン



(たぶん) 世界最後の 8bit パソコン、PC-G850VS を使います。

# スペック

CPU	Z80 相当 CMOS 互換品 @ 8MHz
メモリ	RAM 32KB (約 30KB が使用可能)
ストレージ	RAM と共通。増設不可
画面	144x48 ドット / 24x6 文字 モノクロ液晶
対応言語	BASIC、C、CASL、Z80 アセンブリ

高クロックとグラフィック性能が光ります。電池4本で70時間駆



# 実装の方針

さすがにポケコン単体で TCP/IP を実装するのはつらいので、外部に通信用のコンピュータを接続して、そこで API との通信をおこない、結果をポケコンに転送 & 表示することにします。



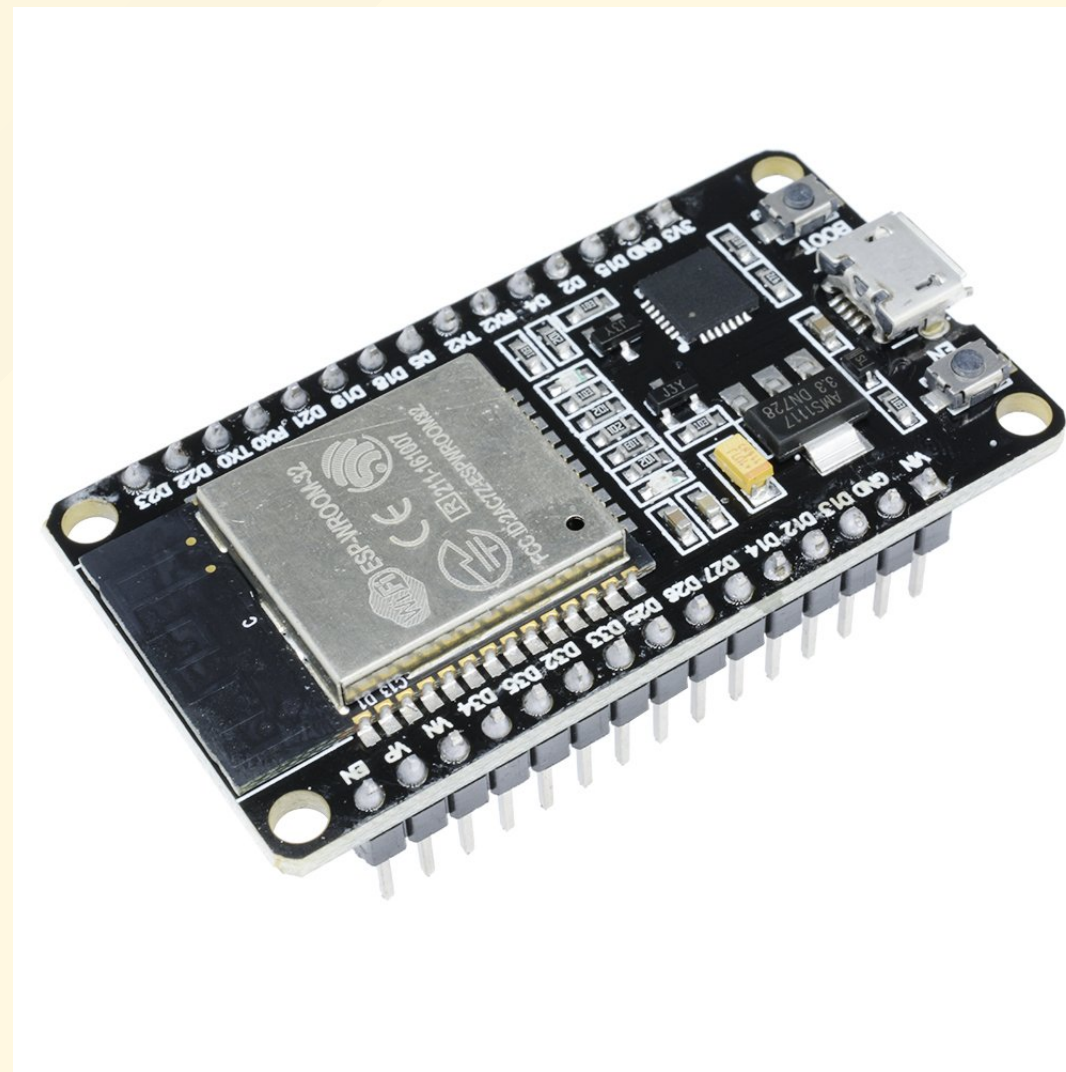
Twitter をしよう！

# 通信用コンピュータ

ESP32 を使います。

AliEx で 700 円位で売ってる古い  
リビジョンで十分です。

Raspberry Pi なんかの Linux ボード  
が使えれば楽ですが、サイズと  
通信規格の問題で採用できません  
でした。





# ポケコン - ESP32 間通信

ポケコン内蔵の Z80 SIO が使えます。

一般的な UART と論理が反対ですが、ESP でソフトウェア的に対応ができます。

ソフトウェアで対処できないときは、NOT  
ゲート  
を 2 個使って、回路で対応しましょう。

最低 3 本線をつなげば、動作できます。

良い子の皆さんは保護回路を入れてください...

# ポケコン側プログラム

BASIC で書くのですが、エラーハンドリングの方法がないので、**完全にエラーが発生しない**ように書かなくてはなりません。

1. バッファをオーバーフローさせない
2. バッファが空のときは読み出さない
3. 通信エラーは起こさない（！）

上記を常に意識していないと、プログラム全体の実行が止まります。

# 漢字を出す

漢字は出ません。

8x8 ドットの漢字フォントを ESP32 のフラッシュメモリに載せて、ビットマップとして転送 & 描画することにしましょう。

高価なフォントデータは旧来 ROM として別売品で売られることが

多かったのですが、もはや ROM でなくなった現在も *漢字 ROM* とよびます。大して役に立たない知識ではあります……

# ひらがなを出す

実はひらがなも出ません。

高校生のころ手打ちしたひらがなフォントデータがあるので、これを流用しましょう。

漢字同様、ビットマップとして描画することになりますが、フォントデータは 600B ほどなのでポケコンに内蔵できます。

2445465C44  
3F40404030  
025A3F0202  
023F424F42  
  
082D5B4908  
720F224A48  
424A4E4B32  
014141413E  
0119254341  
  
314A444241  
0A27523922  
3F40214142  
334C344876  
247F0C6478

# 漢字を入力する

漢字が出ないなら、当然入力もできません。

IME を自力開発する余裕はないので、巨人の肩に乗りましょう。

Google 日本語入力の API があるので、ここを叩いてあげればよさそうです。

しかしポケコンで入力するのは SJIS の半角カタカナ。

API に投げるには UTF-8 の全角ひらがなでなくてははいけません。



ADDRESS	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789
0000000	20	E3	80	82	E3	80	8C	E3	80	8D	E3	80	81	E3	83	BB	縹ゆ・後・阪
0000010	E3	83	B2	E3	82	A1	E3	82	A3	E3	82	A5	E3	82	A7	E3	縹係縹。縹縹縹
0000020	82	A9	E3	83	A3	E3	83	A5	E3	83	A7	E3	83	83	E3	83	縹縹縹縹・縹縹
0000030	BC	E3	82	A2	E3	82	A4	E3	82	A6	E3	82	A8	E3	82	AA	縹「縹、縹縹縹
0000040	E3	82	AB	E3	82	AD	E3	82	AF	E3	82	B1	E3	82	B3	E3	縹縹縹縹縹縹縹
0000050	82	B5	E3	82	B7	E3	82	B9	E3	82	BB	E3	82	BD	E3	82	縹縹縹縹縹縹縹
0000060	BF	E3	83	81	E3	83	84	E3	83	85	E3	83	88	E3	83	8A	縹縹ヤ縹・
0000070	E3	83	8B	E3	83	8C	E3	83	8D	E3	83	8E	E3	83	8F	E3	ル縹後口縹
0000080	83	92	E3	83	95	E3	83	99	E3	83	9D	E3	83	9E	E3	83	偵カ縹空・
0000090	9F	E3	83	A0	E3	83	A1	E3	83	A2	E3	83	A4	E3	83	A6	縹。縹。縹「縹
00000A0	E3	83	A8	E3	83	A9	E3	83	AA	E3	83	AB	E3	83	AC	E3	縹縹縹縹縹縹縹
00000B0	83	AD	E3	83	AF	E3	83	B3	E3	82	9B	E3	82	9C	0D	0A	縹縹縹縹縹縹縹
00000C0	20	EF	BC	8E	EF	BC	BB	EF	BC	BD	EF	BC	8C	E2	80	A6	・趣沙・入
00000D0	E3	82	92	E3	81	81	E3	81	83	E3	81	85	E3	81	87	E3	縹偵＝縹・
00000E0	81	89	E3	82	83	E3	82	85	E3	82	87	E3	81	A3	E3	83	♂縹・e縹
00000F0	E3	81	85	E3	81	85	E3	81	85	E3	81	81	E3	81	83	E3	縹縹縹縹縹縹縹

Twitter をしよう！

# 自作

～ないものは作る～

# UTF-8 についてざっくりと

ASCII 由来の半角英数字と一部記号 (0x00 ~ 0x7F) については、

UTF-8 と SJIS はコンパチです。

が、それ以降はマルチバイトで文字を表します。

具体的には、日本語なら 1 文字あたり 3 バイト。  
絵文字だと 4 バイトになります。

気になった人は**コード表**を見てみよう！

# SJIS についてざっくりと

ASCII との互換、半角カタカナの JIS（旧規格）との互換を保ちつつ、使われていなかった領域（0x80 ~ 0x9F、0xE0 ~ 0xFF）を使って、半角文字は 1 バイト、全角文字は 2 バイトで表す方式です。

ポケコンの文字コードは、「漢字の出せない JIS」といったところ。

漢字領域を使うと文字化けしますが、その他については SJIS と互換します。

# くらべると

	ポケコン	SJIS	UTF-8
半角英数	ASCII 準拠	ASCII 準拠	ASCII 準拠
半角か	SJIS 互換	1 バイト	3 バイト
半角かな	半角か使用	-	-
全角	-	2 バイト	3 バイト

# `iconv` は偉大！

結局、UTF-8 全角 → SJIS 全角は OSS のものを流用。  
そのほかは変換テーブルを自作しました。

PHP だと 1 行で書けるのに（ブツブツ……）



# 出力するには.....

例： 「漢字カナ混じりの文字列を転送」

モード	G	T	G	H	G	H	G	T
データ	漢字	か	混	ジリ	文字列	ヲ	転送	END

- G .....16 進数文字列を送信。ビットマップとして描画。
- T .....テキストを送信。そのまま出力。コマンドもこのモード。
- H .....テキストを送信。ひらがなで出力。

# ポケコン側は.....

モード	G	T	G	H	G	H	G	T
データ	漢字	か	混	ジリノ	文字列	ヲ	転送	END

この入力を受け取り、モードを切り替えながら、画面の左上から順に、出力位置が破綻しないように描画します。

失敗すると漢字がぶつ切りになったり、漢字の表示途中で改行したり

半角文字を漢字で上書きしてしまったりします。

すでにお腹いっぱいかもしれませんが  
これから Twitter と通信します

# PlatformIO はいいぞ

ESP32 のプログラムには Arduino C++を使いますが、さすがに規模が大きすぎて、Arduino IDE で開発するとつらいです。

そこでおすすめなのがPlatformIO！

純粋な C++で開発ができ、VSCode で拡張機能を使うとゴリゴリに IntelliSense がききます。

組み込みマイコン開発の強い味方です。

参考：マイコン開発するなら PlatformIO がめちゃくちゃいい

# OAuth を通す

Twitter に書き込むようなリクエストを発行するときは、OAuth 認証が必要です。普段はSDKを使うと気にしなくてよいところも、もちろん自力実装です。

面倒なのが、**シグネチャの作成**のために SHA1 が必要なこと。組み込み用の `mbedtls` が動いてくれたので、これを使います。

参考：[Twitter API から Arduino – ESP32 を使ってトレンドツイトを取得してみた](#)



# 使用ライブラリとメモリのお話

ESP32 は、実に 320KB もの広大な RAM を持っていますが、通信系の処理がかなり大変らしく、結構ギリギリになります。

その結果、既製品の HTTP ライブラリや JSON ライブラリを使うとメモリが足りなくて動作しなくなっていました。

仕方ないので、最小限必要となる機能だけを実現する、省メモリなライブラリを自作します。

# HTTPS を喋る

ESP32 の `WiFiClientSecure` で、SSL 接続ができます。  
その後の HTTP は自力で喋ります。

自力で HTTP を喋るの、そこそこ楽しいです。

```
117   ...String content = req->bodyParam.get(), header;  
118   ...header = req->getRequestString();  
119   ...header += "Accept-Charset: UTF-8\r\n";  
120   ...header += "Accept-Language: ja,en\r\n";  
121   ...header += "Authorization: " + OAuth_header + "\r\n";  
122   ...header += "Connection: close\r\n";  
123   ...header += "Content-Length: " + String(content.length()) + "\r\n";  
124   ...header += "Content-Type: application/x-www-form-urlencoded\r\n";  
125   ...header += "Host: " + req->host + "\r\n\r\n";  
126
```

# レスポンスをパースする

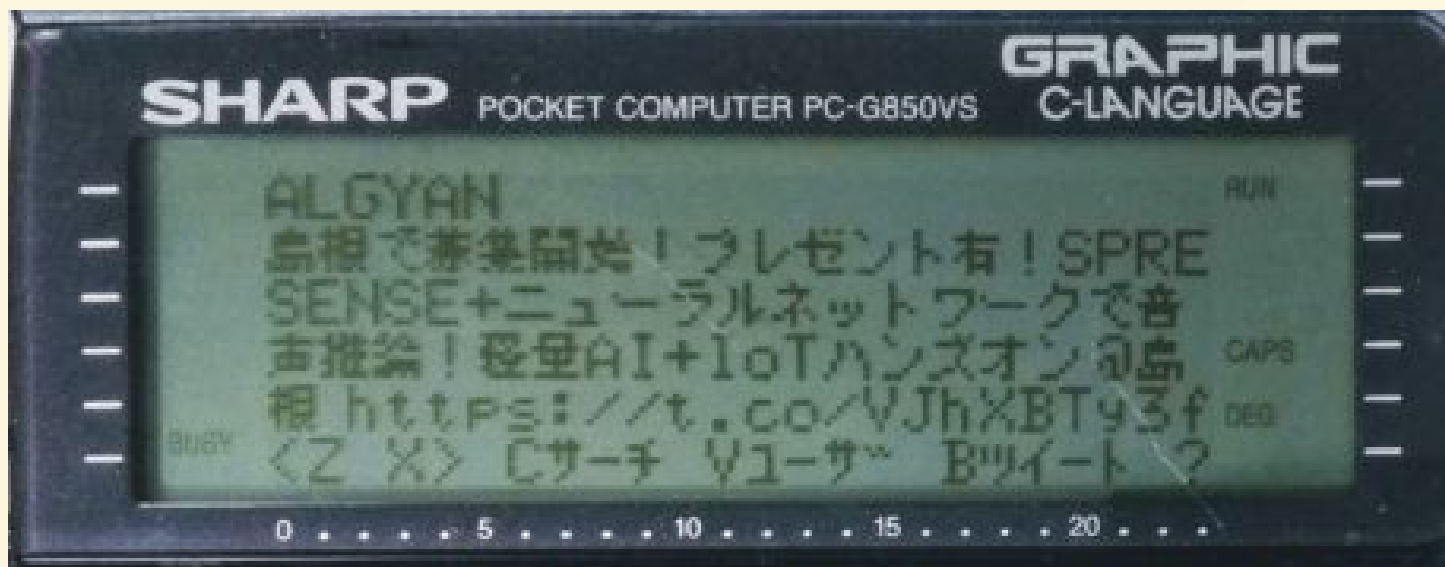
Twitter API からのレスポンスは JSON で返ってきます。

が、JSON はファイルの末尾まで読まないで構造が確定できず、パースする前に全部メモリ上に読み込まないといけません。

そんなことをするメモリはないので、送られてきた JSON を逐次解析しながら、不要な部分は捨ててしまうようにします。

# パースしたレスポンスを転送する

前半でお話しした漢字転送プロトコルでレスポンスを転送すると.....



実際の描画速度が見たい方は [動画](#) をどうぞ。

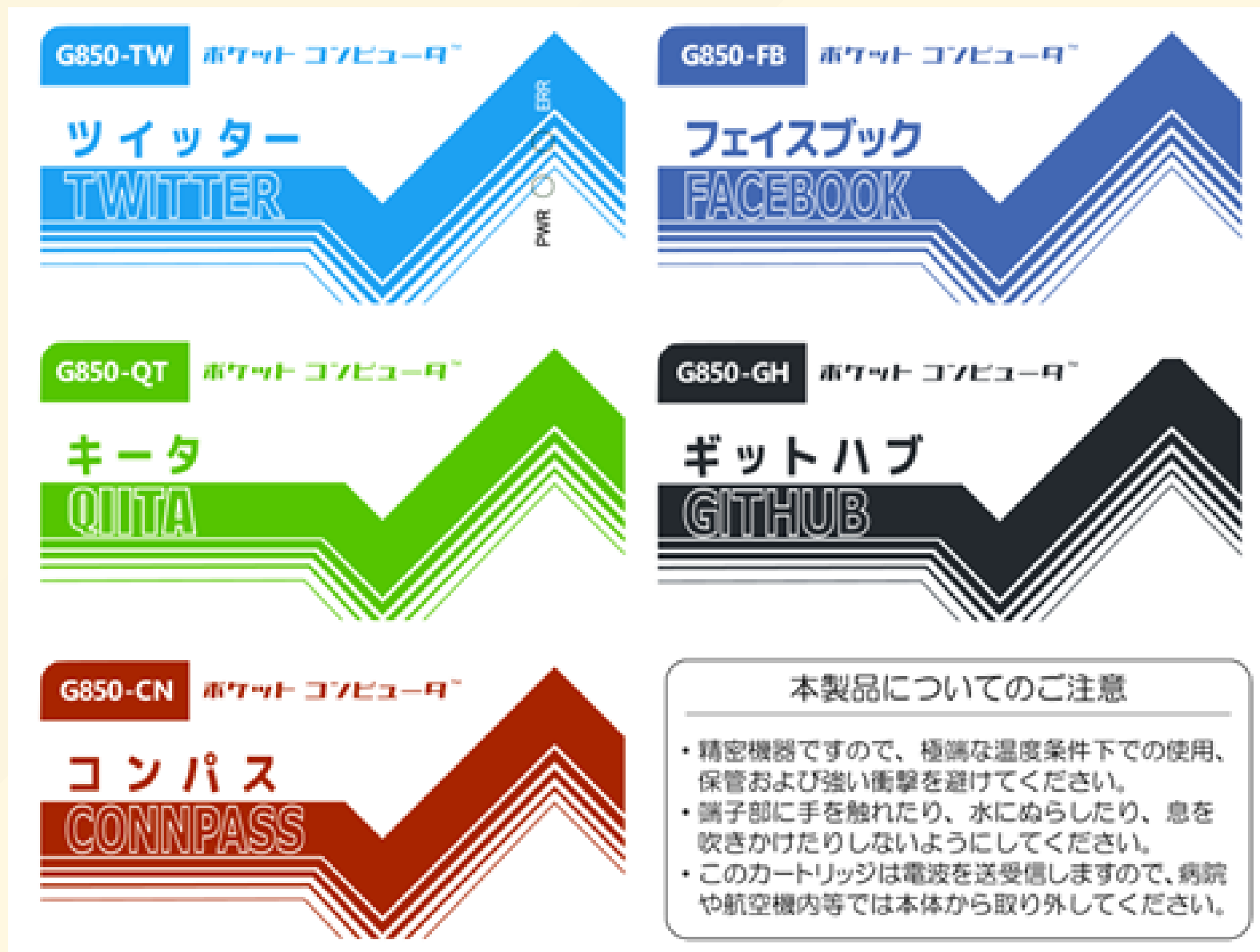
# 所見

- Twitter ができるようになるまで 2 年かかった
- Twitter ができるってすごい
- 漢字が出ることのありがたみ
- 一生 Unicode についていきます
- コンピュータの歴史を切り開いた先人に感謝
- Twitter をやると情報通信技術のアレコレに触れられる



# 今後

姉妹品にご期待  
ください  
(出るのか?)



# おまけ

今回、ESP32 から Twitter に直接アクセスしました。  
ご想像のとおり、SDKを使わないのは茨の道でしかありません。

ただ ESP32 から Tweet したいだけなら、  
SDK で踏み台を作って、HTTP か何かで叩くのがよいでしょう。

勉強のために作るとものすごくためになると思いますが、  
**間違っても本番環境・実プロダクトで真似しないでください  
ね！**

# 君もポケコン、ゲットだぜ！

詳しくは.....

Qiita: [ポケットコンピュータPC-G850VSからTweetする](#)

Note: [ポケコン哀歌、きみと一緒に歩いた一年。.....ポエム](#)

GitHub: [plageoj/twitter-for-pc-g850](#) ...ESP32側

Gist: [plageoj/N.F](#) & [plageoj/TW.BAS](#) ...ポケコン側