

# A Dive to Ada

– My first step to programming in Ada–

## pigpio とは

pigpio は、手のひらサイズのワンボードコンピュータ Raspberry Pi のハードウェアを制御するためのライブラリで、C 言語で書かれています。これを使えば汎用 IO (GPIO) やシリアル、I2C、SPI などの各種インターフェイスをユーザープログラムから簡単に利用することができます。同種のライブラリとしては Wiring Pi があり、様々なプロジェクトで使われていますが、pigpio はより多くの機能があり、なんといっても Raspberry Pi 財団が公式にサポートする Debian GNU/Linux ベースの OS である Raspbian では Jessie (Old Stable, 2017 年 12 月 28 日時点での Stable は Stretch) からはバイナリパッケージが提供されていて、インストールが手軽なのもメリットです (Wiring Pi はソースを git clone で持ってきて自分でビルドする必要があるのでもっとだけ面倒です。ビルドの手順そのものは難しくありません)。

The pigpio library : <http://abyz.me.uk/rpi/pigpio/index.html>

## Ada とは

軍用組み込み機器のプログラム作成のために開発された (といっても、今は最近の一部の軍用機向けのソフトは C++ で書かれているものもあるらしいですが) 手続き型の言語で、型付けがとても厳しく、文字列処理に関しては固定長文字列型の存在 (バッファオーバーランを防ぐため?) と、それとは別に可変長文字列型 (上限付きのものと、上限のないものの 2 種) があること、その他 C# や Visual BASIC のプロパティにちょっと似た感じの「アトリビュート」といったユニークな機能をもつ一方で、C のポインタに相当するもの (関数へのポインタも使える)、演算子や関数、手続きのオーバーロードや、Ada 95 という規格からはオブジェクト指向が導入されるなど「今風」の機能ももちろんあります (最新の規格は Ada 2012)。フリーの Ada コンパイラである GNAT (gcc の一部) もあって、Raspbian にも gnat というパッケージが提供されているので、Ada 2012 の言語仕様の Ada コンパイラが apt-get 一発でインストールできます。ちなみに、ソースコードの見た目は Pascal で、機能的にも「Pascal の親玉」的な雰囲気です。

Ada プログラミングに関しては、日本語の文献や紹介サイトはあまり見かけませんが、英文だと結構あります。独学する上で個人的に参考になったサイトをいくつか挙げておきます。

- [https://gcc.gnu.org/onlinedocs/gnat\\_ugn/Getting-Started-with-GNAT.html#Getting-Started-with-GNAT](https://gcc.gnu.org/onlinedocs/gnat_ugn/Getting-Started-with-GNAT.html#Getting-Started-with-GNAT)

- [https://en.wikibooks.org/wiki/Ada\\_Programming/All\\_Chapters#.22Hello.2C\\_world.21.22\\_programs](https://en.wikibooks.org/wiki/Ada_Programming/All_Chapters#.22Hello.2C_world.21.22_programs)
- <https://www.dwheeler.com/lovelace/master.htm>

Stack Overflow でも Ada 関係の質問が出ているようです。海外では思っているより普通に使われているのかもしれませんが。

## Pigpio\_Ada とは

上記の pigpio ライブラリを Ada から使えるようにするための wrapper です。自分で使いたかったのと、Ada プログラミング、とりわけ C で書かれたプログラムとの連携方法の勉強のために作ったため、まだ pigpio のごく一部の機能しか使えません。ソースコード等は GitHub で公開します(反響があれば開発を続けていく…かもしれません)。コミックマーケット 91 (昨年の「冬コミ」) 以降頒布した当サークルの同人誌関係のファイルもありますので、そちらもよろしければどうぞ。

Mercury 日本橋 同人誌サポート用レポジトリ：

<https://github.com/jalumi/doujinshi>

現状で使える Pigpio\_Ada の関数・手続きは以下の通りです。名前は pigpio の関数の名前の先頭に” Pi” をつけたものです。機能の詳細な説明や引数の意味は pigpio のサイトの “pigpio C Interface” のページを参考にしてください。

pigpio C Interface：<http://abyz.me.uk/rpi/pigpio/cif.html>

Essential	
procedure PiGpioInitialise;	ライブラリの初期化
procedure PiGpioInitialise;	ライブラリ利用終了
For Beginner	
procedure PiGpioSetMode ( gpio : in GPIO_Range; mode : in GPIO_Mode);	指定した GPIO のモードを設定する
function PiGpioGetMode ( gpio : GPIO_Range) return GPIO_Mode;	指定した GPIO に設定されているモードを取得する
procedure PiGpioSetPullUpDown( gpio : in GPIO_Range; pud : in GPIO_PUD);	指定した GPIO のプルアップ／プルダウンを選択する

function PiGpioRead ( gpio : GPIO_Range) return GPIO_Level;	指定した GPIO の状態を読み出す
procedure PiGpioWrite ( gpio : in GPIO_Range; level : in GPIO_Level)	指定した GPIO に書き込む
Serial	
function PiSerOpen( sertty : in String; baud : in SpeedSelection; serFlags : in Natural := 0) return Handle	シリアルポートを開く
procedure PiSerClose( hdl : in Handle)	シリアルポートを閉じる
procedure PiSerWriteByte( hdl: in Handle; value: in UByte)	シリアルポートに 1 バイト書き込む
PiSerReadByte(hdl : Handle) return UByte;	シリアルポートから 1 バイト読み出す
Utilities	
function PiGpioVersion return Positive	pigpio のバージョンを得る
procedure PiGpioSleep( timetype : in Time_Type; seconds : in Natural; micros : in Natural)	指定した時間だけ待つ

## Pigpio\_Ada を使ったサンプルプログラム

Ada で L チカするのもそれなりに達成感があるのですが、もう少し規模が大きくて、Ada のプログラミングや、wrapper の使い勝手が評価できるようなものとして、古き良き時代の CASIO のゲーム電卓で遊べた「デジタルインベーダー」もどき “*Digital InvAda*” (invada はルーマニア語で invade の意味らしいです) を作ってみました (数字の「インベーダー」が右端から左方向に攻めてきて、自分の砲台の数字インベーダーの数字に合わせて攻撃する、というアレです)。うまくいけば動く状態のものをサークルスペースで実演展示する予定です。このサンプルプログラムのソースコードも上記の GitHub のサポートレポジトリで公開予定です。

表 題	<b>A Dive to Ada –My first step to programming in Ada–</b>		
サークル名	Mercury 日本橋		
発 行 者	Kasumi YOSHINO (JA1UMI)	E-mail:	ykasumi@spamex.com
		Twitter:	@yKasumi
		Mastodon:	
			@yKasumi@mastodon.cloud,
			@JA1UMI@hamtter.net
発 行 日	2017 年 12 月 29 日 (コミックマーケット 93) 初版発行		