# 情報

データベース

Phpを使う。

Json文字列を飛ばして、中継サーバーがデータベースに書き込む

python -m SimpleHTTPServer

# 全般情報

仕様箇条書き：

・マイコンの起動後にReady信号のようなものが必要。

## マイコン

とりあえずまとまりもなく箇条書きにする。

・マイコン内には大体複数個の発信子があるので、それを選んだり、動作周期を選択したり、と初期設定が必要

・開発時にはエミュレータを用いる。（毎回ROMに書き込むとかできるわけない）

・機械制御で扱うような周期の場合は、マイコンレベルでは気にする必要がほとんどない。

電圧が高い方がHIGH になるまでの時間が長く、マイコン内と制御機器系統では電圧が桁レベルで異なるので当然ともいえる。

・CPUは演算のみ。マイコンはRAM（キャッシュ）や発信子を含んだIC。

・オンチップ発振子はCR（コンデンサと抵抗）などで，水晶などよりもはるかに精度が悪い。

・CPUクロックは通常，どれだけ細かくても発振子の1/2の周波数になる。電気は完全な矩形波は不可能で，必ずサイン波のようになる。そうするとズレ具合によっては毎回Lowのポイントしか取れなかったりしてしまう。その為、必ず2点で取る必要がある。

蛇足１）高級CPUだと専用回路でクロック信号を反転させ，生の信号と併用する事で信号の信頼性を得ているものもあるとか。

蛇足２）CPU内部専用で逓倍した高速クロックを利用したりするが，高速である為に非常に小さな電圧値となり，とても外部では使えない。

蛇足３）ちなみに通信の場合は，たとえば10分割した１波の信号から奇数個のポイントで多数決を取ってHighかLowかを判断するものがあるらしい

蛇足４）マイコン内部で使っている数種類のクロックは，互いに若干のズレがある。（流れる電流量を抑えたり，処理の高速化などに使われる）

・時計クロック

省エネ用クロック。wait状態（＝パソコンのスリープ状態）時に使用したりする。大体のマイコンには付いている。

低速である為に省エネで，外部からの割り込みが入ったら高速のクロックに切り替えたりする。

・ウォッチドッグ(WDT) は通常，メインのタイマと関連のある周期にしておく。

ウォッチドッグは不用意な割り算などで，設計通りの時間内に１サイクルが終わらなかった時などに作動する。

逆に言うなら，問題なく１サイクルを終えた時はWDTをクリアする必要がある。

・メインタイマより更に細かいタイマを動作させてカウントアップさせる，わざわざマッチコンペアを使わなくても似たような事ができる。また，速さが求められる処理はそのタイマーの割り込みを使ったりして調整ができる。

本流はあえて低速なタイマーを動作させる，という設計。

・あるタイミングを待つためにはNOOP(C言語ではWait) を用いれば良い。

・マイコンの寿命は約10年

## マイコンプログラミングの基本

ROMにはプログラムを置く。C言語は「コンパイルされたマシン語は実行中に変える事がない」前提で作られているので、ROMとの相性が良い。

RAMにはスタック領域（関数ポインタなど），ヒープ領域（動的領域）、静的領域（グローバル変数）など、変数が割り当てられる。

例外的にプログラムがここに置かれる事もあるらしい。

参考）[メモリの領域](#_メモリの領域)

## PWMボード

Arduino＋PCA9685 ※またはラズパイでも良し。

NXPと言うところが出しているPWMコントローラ「PCA9685」を使った基板を用いる。

<http://akizukidenshi.com/catalog/g/gK-10350/>　他にも２、３社似たようなのを出している。

長所：メモリに値を書き込むか、python使って値だけ送れば良い

短所：そもそもPCA9685自体が廃盤になったら対応できない。

# 三菱サーボドライバの情報

## パルス列信号の仕様

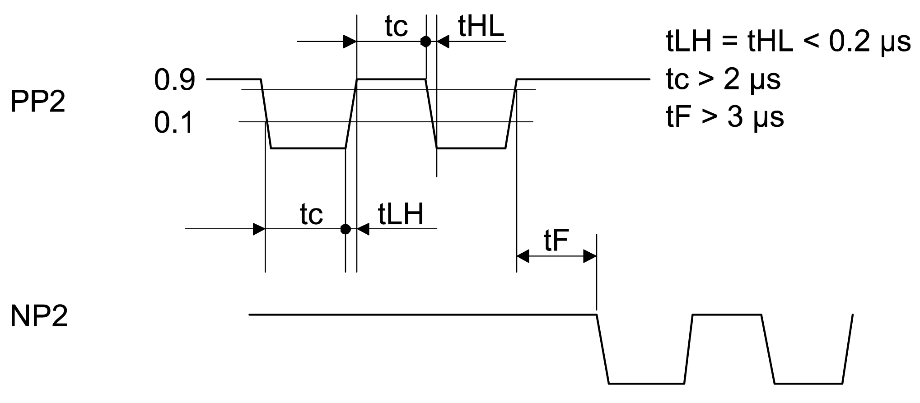
MR-J4\_A サーボアンプ技術資料より

入力電圧24V

サーボドライバ内部には1.2kΩの抵抗が付いている。つまりドライバ内部のフォトカプラをonさせる為に20mA流す必要がある，と言う事。

VCES ≦　1.0V 要するにLOW時の電圧を1V以下にしろ，と言う事？

ICEO ≦ 100μA 上に同じく



0.2μSec = 500kHz

※tHL の長さがよくわからないが？

# 詳細情報

## マイコン情報

### ラズパイ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | プロセッサ | クロック | オシレータ | 価格 |  |
| 4+ | ARM Cortex-A72 | 1.5GHz |  | \5,225 | 7,700(4GB RAM) |
| 3B+ | ARM Cortex-A53 | 1.4GHz | 19.2MHz | \3,978 |  |
| Zero | ARM1176JZF-S | 1.0GHz |  | \2,200 |  |

開発環境

Linux OS(Raspbian) 上で動くpython。ただしラズパイにはハードウェアクロックもある。

Web上ではC言語による情報もあるが、詳細は不明。

microPython というものを使うと、ベアメタルに近い開発も可能。但し資料が圧倒的に少ない。

### Arduino

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | プロセッサ | クロック |  | 価格 |  |
| Uno R3 | ATmega32 8p | 16MHz |  | \3,300 | PWN pin6本 |
| ESP32 | Tensilica LX6 | 32kHz |  |  | Arduino core for the ESP32 |

開発環境

独自（Arduino IDE） C/C++ を使う。 オープンソース

Eagleファイル：（CC-BY-SA）著作権表示、元作品のライセンスを継承。商用利用は可能。

Arduino　IDE（GPLライセンス）

<https://www.arduino.cc/en/Main/CopyrightNotice>

### ルネサス

RXxxT シリーズがモーター向きとか。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | クロック | 電源 | 値段 | 備考 |
| RX72T  32bit(RXv3) | 外部最大  200MHz | 2.7-5.5V | \12,200？ | 5相モーターの制御も可能  オンチップエミュレータ  タイマ：（コンペア）  TMR0×8 コンペア対応　比較レジスタ２つ  　CMT×4　×比較レジスタ２つ  　※但しコンペアマッチタイマはペアで使う |
| RX66T  32bit(RXv3) | 外部最大  160MHz  オンチップ  20MHz | 2.7-5.5V | \8,800(100B) | タイマ：（コンペア）  TMR0×8 コンペア対応  CMT×4　×比較レジスタ２つ  ※但しコンペアマッチタイマはペアで使う |
| RL78/G14  16bit | 外部最大  20MHz  オンチップ  32MHz | 1.6-5.5V  VDD  (FETのプラス) | \6,500(100B) | ステップモーター向き推奨　ポリテク講習。RL  タイマ：（コンペア）  RG ×比較レジスタ４つ　比較レジスタ２つ  RD(TRD0, TRD1) ×比較レジスタ４つ  RJパルス出力 |
| RX24T  32bit (RXv2) | 外部最大  80MHz  オンチップ  64MHz | 2.7-5V | \5,500(100) | タイマ： （コンペア）  TMR×8 コンペア対応　比較レジスタ２つ  CMT×4　×比較レジスタ２つ  ※但しコンペアマッチタイマはペアで使う |

値段は開発ボードのもの。RL, RX, RA, RZ とあり、右に行くほど上位

E2エミュレータ―が高い \66,000 E2 lite　だと\7,980：[対応] RL78, RXシリーズ

E8a：[対応] R8C, R32C, H8S/Tiny など

E1：[対応] RX, RL *2019年12月31を最終オーダーに生産中止*

5or3.3V電源供給。

E2 Lite：[対応] RX　大体\8,000程度

対応（RX66T）

E2：[対応] RX, RL

E2Liteに加え、「ホットプラグイン機能，外部トリガ入出力」などが可能

E20：[対応] RX600, RL 大体18万

E1＋大容量トレース。但しエミュレータからの電源供給は非対応。

HEW

CS+（CubeSuite+）

e2studio

ESP8266

MicroPythonの本で使われているマイコン。

開発環境

Eclipseベースの独自IDE（e2Studio）。C/C++。

無料評価版あり。professional版を60日間使える。期間後はstandard版（リンクサイズ制限有り）になる。

ヨーロッパ製らしい。多分ちょっと使いにくい。pro版は20万くらいするらしい。（情報少ない）

CubeSuite+(CS+) 　少し古い開発環境で、20万くらいするらしい。

STMicroelectronics(STMicro)

### 推奨ルネサスデバイスマイクロコンピュータ

|  |  |  |
| --- | --- | --- |
| ブロック | 推奨品名 | 特長他 |
| サーボ  制御用MCU | RZ/T1  \8,603 | 動作周波数 (MHz) : 600/450/300  動作電圧 (V) : 3.3(I/O), 1.2(Core)  搭載メモリ (max.) : 密集合メモリ 512KB+32KB, 拡張1Mbyte RAM  密結合メモリによる高速リアルタイム制御  R-INエンジン内蔵による高速・低消費電力通信  エンコーダI/F搭載による外部FPGA 機能の取り込み |
| RX72M  \28,408 | 動作周波数 (MHz) : 240  動作電圧 (V) : 3.3  搭載メモリ (max.) : 4MB Flash, 1MB RAM, 32KB Data Flash  EtherCAT slave Controller 搭載により通信とモータ制御を１chipで実現 |
| RX71M  \15,134 | 動作周波数 (MHz) : 240  動作電圧 (V) : 3.3  搭載メモリ (max.) : 4MB Flash, 522KB RAM, 64KB Data Flash  高性能RXv2コアと大容量メモリにより高精度なモータ制御を実現 |
| RX72T  \3,425 | 動作周波数 (MHz) : 200  動作電圧 (V) : 2.7 to 5.5  搭載メモリ (max.) : 1MB Flash, 128KB RAM, 32KB Data Flash  RX66Tの高性能版：200MHz化、RXv3レジスタ一括退避機能、三角関数演算器搭載 |
| RX66T  \3,696 | 動作周波数 (MHz) : 160  動作電圧 (V) : 2.7 to 5.5  搭載メモリ (max.) : 512KB Flash, 128KB RAM, 32KB Data Flash  高性能RXv3コアと多様なモータ制御機能を搭載 |

<https://www.renesas.com/jp/ja/solutions/industrial-automation/motion-and-drives/ac-servo.html>

## マイコン詳細情報

### まとめ

特徴

ラズパイ Etherが強く、サーバーとしても使える。OSが載っているので起動や動作にやや問題。

Arduirno Etherなし，比較的純粋なマイコンタイプ。

Uno Rev3（ボード）はマイコンにATmega328を使用。

PIC, AVR(ATMEL) マイコン単体は安いが，ライターがやや高い。趣味でたくさん作る人むけ。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | RX72T | ESP32 | micro:bit |  |
| 動作電圧[V] | 2.7～5.5 | -0.3～3.6 | 1.8～3.6 |  |
| 推奨電圧[V] | 5/3.3 | 3.3 | 3.0 |  |
| 絶対定格[V] | 6.5 | 3.6 | 3.9 |  |
| IO上限 [A] | ？ | 1200mA |  | 電源の性能による。 |
| IO絶対 [V] | VCC+0.3 |  | VDD+0.3 |  |
|  |  |  |  |  |

### ラズパイ

#### 3B+

##### 概要

MPC：Broadcom BCM2837B0 コア：ARM Cortex-A53 (ARM v7)　上限周波数がどこにも書いてない

CPUクロック 1.4GHz

タイマー 64bit

ARM Timer 64bitタイマー×１（オンチップのタイマーと言う事？）

System Timer ×１

Compare が４つあるがC0とC2はGPUで使用しているので，有効なのはC1とC3だけ。

※よくWeb上に「システムタイマーが４つ」という情報があるが，あるのは４つのCompare

アドレスはBroadcomのPeripheralsのp172に、0x7E003000を始めとしたオフセット形式で記述がある。

（マニュアル）<https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2837/README.md>

プリスケーラ― 32-bit

<https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2837b0/README.md>

外部発振子 19.2MHz ？？？Cortex-A53の限界が明記されていないが、なぜこんなに高速？

GPCLK tJOSC Osillator-derived GPCLK 20ps

GPCLK tJPLL PLL-derived GPCLK 48ps

英語だが、興味深い情報ソース

<https://www.tablix.org/~avian/blog/archives/2018/02/notes_on_the_general_purpose_clock_on_bcm2835/>

その派生ソース（日本語）<https://blog.boochow.com/article/raspberry-pi-hw-pwm-driver-1.html>

アドレスマップ

0x4000\_000～0x4001\_FFFF ARMタイマー，割込み

4B

MPC：Broadcom BCM2711 コア：ARM Cortex-A72

クロック 1.5GHz

オシレータ

<https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/README.md>

Quad core Cortex-A72 (ARM v8)

### ルネサス

メモ：公式から「ドキュメント → ドキュメントタイプ：ユーザーズマニュアル：ハードウェア」で日本語版のマニュアルが手に入る。

RX72T [32bit(RXv3)] 外部最大200MHz \12,200(144pin)

RX24T [32bit (RXv2)] 外部最大80MHz \5,500(100 pin)

タイマ 【 RXシリーズはこのパターンが多い 】

TMR0×8 コンペア対応 ×比較レジスタ２つ（コンペアマッチA，Bの２種類）

CMT×4　×比較レジスタ２つ（コンペアマッチA，Bの２種類）

※CMTでコンペア使うときは32bitモードのみ（２つのタイマを１対で使う）

クロック分周：1,2,4,8,16,32,64

### Arduino

Arduino Uno Rev3

マイコンがDIP型で取り外しできる。書き込んだら別の回路で使用可能。

（但し取り換えるマイコンにブートローダを書き込む必要がある；書き込まれたものも販売している）

動作電圧：5V （5V限定らしい）

クロック：16MHz（水晶）

MPU：ATmega328P

内部RCオシレータ：8MHz

8bitタイマー（２つ）：

16bitタイマー（１つ）：

共通：

・10bitプリスケーラ

・各タイマーにCompare Match ２つずつ。

### ESP

（CPUはTensilicaのXtensa LX6だが，マイコンは上海TSMCが開発）

<http://akizukidenshi.com/catalog/g/gM-11819/> \1,280（ESP-WROOM-32）

<https://www.switch-science.com/catalog/3210/> \2,200（ESPr Developer32）

PWM 独立16チャンネル

80MHzのAPBクロックを使用(16bitカウンタ) Advanced Preipheral Bus ⇐ 何の事？

8MHzオシレータを使うオプションがある（？）

カウンタを調整する事で40MHz までのPWMに対応できる。

パルスカウンタ 8チャンネルある。設定値に到達すると割り込みが入る。

１チャンネルに付き、一度に４信号をキャプチャできる。

※7つのモードを通して　という表現があるが、何のことか分からなかった。

ESP8266 micropython

<http://akizukidenshi.com/catalog/g/gK-12236/> \1,280（ESP-WROOM-002）

PWM PWM用途に使えるgpioが4つある。100Hz～1kHz

refresh rate が14bit ⇒14bit カウンタという事？

PIC12F1572 \80

開発ボードはない？ pinの数が8つ。

動作周波数：最大32MHz

タイマー

8bit×2　TMR0(8bit プリスケーラ), TMR2(PR2と比較)

16bit×1 TMR1

### MicroPython

・マイコン用のコンパイラで、Python3とかなりの互換性をもつ。低レイヤ向きのモジュールも含まれる。

・ソースコードはGitHub上でMIT Licenseで公開。

必須：著作権の表示 許可：商用利用、修正、配布、サブライセンス 禁止：責任免除

・日本語での情報が限りなく少ない。

・ARM Cortex-M、ESP8266、ESP32、16bit PIC、Unix、Windows、Zephyr　をサポート

ボード

MicroPython pyboard

もともと開発の際に使われたボードがpyboardらしい。

#### 始め方

PCからusbで電源は供給される。

参考資料）ESP32 での MicroPython の始め方

<https://micropython-docs-ja.readthedocs.io/ja/latest/esp32/tutorial/intro.html#esp32-intro>

#### microbit

16bit PWM

イギリスのBBCが主体となって作った教育向けのマイコンボード。CPUはARM Cortex M0

<https://sanuki-tech.net/micro-bit/micropython/about-micropython/>

\3,300

ボタンスイッチ × 2個 加速度センサ × 1個 磁力センサ × 1個

ポートは４つに見えるが、アナログ３つ、デジタル１７有る。I2C, SPI通信もできる。

Wifiは無いが、ブルートゥースがある。（但しBLE；範囲は5Mほど）

アナログ出力は驚いた事に音声も出せる。

3V電源：単４×３のキットや、ボタン電池のキットがある。 開発時はUSBから給電。

BLEは5mしか使えないらしい。⇒ 逆に近くに居る事を知らせる為には使える

リモコンなんかの用途で使えるかも？

チュートリアルが公開されている：

<https://microbit-micropython.readthedocs.io/ja/latest/>

電源 1.8-3.6V

ROM 256KB

RAM 16kB

### Etherマイコン（MWL IoT）

ArduinoはあまりEtherには強くない。

ラズパイZero辺りが一番良い。

## サーボ制御基板(HAT)

SparkFun Pi Servo pHAT

I2C経由で最大16台のサーボと接続？ \1468

電源はこの基板のmicroUSBポートより供給

PCA9685：この基板で使われているPWM/サーボドライバ

25MHz オシレータ

50MHzまでの外部クロックを受け付けるピンがある。

最大1MHz（Fast-mode Plus family）　←何が？

PWM周波数は24Hz～1526Hz まで。

PCA9635という類似品もある　typical 97.6kHz こっちの方が良いのでは？

<https://www.switch-science.com/catalog/3485/>

<https://www.sparkfun.com/products/14328>（製造元：英語）

Adafruit 16チャンネル PWM/サーボ HAT

5V電源をHAT基板に印加する。丸穴のドライバー端子とスクリュー端子台が付いている。

PWM：1.6kHz 12bit 1bitなら6MHz と言う事？？？

**※結局SparkFun と同じPCA9685を使っている**

<https://www.switch-science.com/catalog/2181/>

<https://www.adafruit.com/product/2327> （製造元：英語）

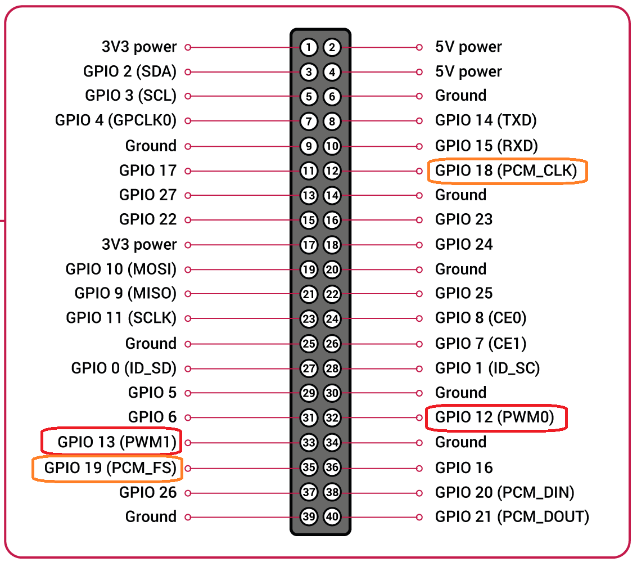
秋月も同じようなものを出している

16チャンネルサーボ制御キット。I2C経由で制御する。

<http://akizukidenshi.com/catalog/g/gK-10350/>

ラズパイ

概要

****ハードPWM

PWM専用のハードウェアがある。

GPIOの12,13,18,19番　ただし独立では２軸。

<https://www.raspberrypi.org/app/uploads/2012/02/BCM2835-ARM-Peripherals.pdf>

pigpioというライブラリを使うらしい。

設定上では1～125MHz周期に設定可能。

1,000,000　を100%に見立ててデューティ比を決める。

・micro python を使うと、いわゆるベアメタル開発ができる。

情報元

<https://qiita.com/s417-lama/items/0ef64a7af3fcf6a56cc5>

出力される周波数 = 19.2MHz / (pwmr \* pwmc)

最大4.8MHzまで行ける？但しPythonでは難しく、ラズパイソフトの限界として300kHzくらいらしい。

<https://teratail.com/questions/193753>

ソフトPWM

LEDの明るさを変えたりするのに使う。

CPUの負荷率でパルス幅が振れる為、精密な処理には使えない。

### ボタンHAT

JAM HAT

ボタン２つとLED6つ。ブザー１つ　\2200

スイッチサイエンス

ボタン５つ \803

<https://www.switch-science.com/catalog/3669/>

## ハード設計メモ

マイコンのポートからの出力で24V系の信号を制御する際、フォトカプラを使う事が多い。

しかし、フォトカプラは大体10～20mA流すので，マイコンにとってはそこそこの電流量となる。

このような状況で極小電流でフォトカプラをON/OFFしたい場合，２重にトランジスタを重ねた「ダーリントントランジスタ」が使われたりする。

<http://sudoteck.way-nifty.com/blog/2010/09/post-07ff.html>

### 酸素濃縮器基板

#### 使用素子

|  |  |  |  |
| --- | --- | --- | --- |
| 型番 | メーカー |  | 備考 |
| G4A-1A-E | omron | パワーリレー |  |
| G3VM-61AR　621(633) | omron | MOS FETリレー | 要するにフォトカプラ |
| TBD62083A | 東芝 | トランジスタアレイ | ２８２００３が有名らしい |
| TC74VHCT541AFK | 東芝 | ロジックIC |  |

# 用語

CCP；Capture Compare

レジスタとコンパレータで構成されていて、文字通り値を取得して比較する。

要するにPWM時の高速パルス数を保証する為の機能。

（どうやらPICマイコン特有の言葉らしい）

CCR；Capture Compare Register

R0, R1, R2

データレジスタ。転送、計算、論理処理に使われるレジスタ。

A0, A1

間接アドレスと相対アドレス用のアドレスレジスタ（意味不明）

FB；Frame Base Register

相対アドレス用のレジスタ

USP；User Stack Pointer

ISP；Interrupt Stack Pointer

割込みベクトルテーブルの開始アドレス

FLG；FLG register

Arduino

Arduinoの特徴

Arduino(アルディーノ)は制御寄りで、PLCをマイコン化したような感じ。

以下にArduinoの特徴を示す：

・開発用には別途PCが必要。Arduino IDEを使う（Win/Mac/Linux対応）

・Arduino専用のC++に近い言語を使う。

・シールド、と呼ばれるセンサ類が豊富に販売されている。

・高速信号に対応しているので、PWM、サーボ運転すら可能。

・基本的にシングルタスク。

・シングルタスクである事もあって、Ether機能はラズパイに比べると弱い。もしくは高値。

・起動は一瞬。

・GPIOは1～17MHz。(0.58～1μSec)。ラズパイはmSec単位（数kHz？情報が少ない）

別売りボード

仕様比較

Arduino – ESP32

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Clock[MHz] | PWM最大 | 周波数 | その他 |
| ESP32 |  |  | 40MHz | 1bitの分解能 ESP-WROOM-32 |
| ESP8266 |  |  | 1MHz | directWrite を使うと良い。 |
| UNO | 16 | 12 | 64kHz |  |
| Due | 84 | 12 | 200khz | CAN, USB |
| Mega2560 | 16 | 48 | 31k～30Hz |  |
|  |  |  |  |  |

※UART, I2C, SPI は標準

シーケンサは標準で100kHz、高速は200kHz

理論値と実値の情報が混同しており、いまいち分かり難い結果になった。

注意点：

例えばMega2560はPWMのピンが15本あると書いてあるが、２、３個のピンでタイマーを共有していたり、16bit timerも限られているため、すべてが利用可能なわけではない。

timer0はシステムで使用しているので使用不可。

ラズパイとArduino 比較

|  |  |  |  |
| --- | --- | --- | --- |
|  | Arduino  ESPr Developer32 | RaspberryPi 3B+ | ルネサス |
| 値段 | \2,200 | \6000 |  |
| 開発環境など |  | MicroPython |  |
| CPUクロック | Tensilica LX6  240MHz | ARM Cortex-A53　1.4GHz |  |
| 参考  最大周波数 | 40MHz | 200kHz |  |
| 独立制御軸数 | 4? | 2 |  |
| 信号電圧 | 2.2～3.6VDC | 3.3VDC |  |
|  |  |  |  |
| 備考 | マイコンらしいマイコン  OSがない  IDEは無料（≒C++） | ソフトウェアPWMとハードウェアがあり、ハードウェアの仕様 |  |

Arduino側情報元

<https://www.mgo-tec.com/blog-entry-ledc-pwm-arduino-esp32.html/2>

仕様：<http://akizukidenshi.com/catalog/g/gM-11819/>

ラズパイ側情報元

<https://teratail.com/questions/193753>

ESP32

ルネサス　RL, RX, RA, RZ とあり、右に行くほど上位

　但し専用IDEが高い。

Armadillo 言い方は悪いがラズパイの劣化版みたいな感じ。