RumiCarインターフェイス資料（第三世代）

2024年6月2日版

目次

[0. 定義・参考情報 1](#_Toc156671353)

[1. 注意 1](#_Toc156671354)

[2. 接続 2](#_Toc156671355)

[3. CM接続 14](#_Toc156671356)

表目次

[表 1　ESP32のUART 3](#_Toc156671495)

[表 2　第三世代　車体-CM接続インターフェイス(EX-IF)(旧表記) 4](#_Toc156671496)

[~~表 3車体-CM接続インターフェイス(縦表記)~~ 5](#_Toc156671497)

[表 4　32型DRV8835結線図 7](#_Toc156671498)

[表 5　Arduino nano(USB端子を下とした図) 9](#_Toc156671499)

[表 6　ESP32/ESP32DevKit　PWM関係ピンアサイン 10](#_Toc156671500)

[表 7　ESP32DevKit接続図(USB端子を下とした図) 10](#_Toc156671501)

[表 8　VL530LXとDRV8835仕様 12](#_Toc156671502)

[表 9　CM用コンピュータ仕様 13](#_Toc156671503)

[表 10　DC-DCコンバータ仕様 14](#_Toc156671504)

[表 11　CM接続インターフェイス 18](#_Toc156671505)

[表 12　I2Cアドレス 19](#_Toc156671506)

図目次

[図 1　24型結線図 6](#_Toc156671510)

[図 2　32型結線図 7](#_Toc156671511)

[図 4 RumiCar インターフェイス 17](#_Toc156671512)

# 定義・参考情報

* 1. 電線
     1. 電力系　AWG24（最大3.5A）程度
     2. 信号系　~~AWG30（最大0.86A）程度~~マルツではAWG28までしか売ってない。AWG28(最大1.2A)
  2. 電源
     1. 車体から供給される電源は基本4.5Vと仮定されて仕様が決められている。これ以外の電圧の車体で利用する場合はCM側の電圧昇降圧機能を鑑みて運用を考えること
  3. CM
     1. コンピュータモジュールの略。ArduinoやESP32が搭載されRumiCarと接続して使われる

# 注意

1. DRV8835の信号線はDRV8835のVCCが上限になる。信号線接続時は接続コンピュータの信号最大値に気を付けること
2. VL53L0Xの信号最大電圧及びI2C信号電圧にも気を付けること。基本的にVL53L0XのVCCと合わせる。
3. Arduino系のPWMピンは3,5,6,9,10,11の６本。タイマ0のみが960Hz、Servoライブライを使うとタイマ1（9,10番ピン）のPWMは使えなくなる。これより操舵を3,11番(タイマ2)、走行を5,6番(タイマ0)、サーボを9,10番(タイマ1)とするのが良いだろう。

表 　ArduinoタイマとPWM

|  |  |  |
| --- | --- | --- |
| タイマ | ピン | 周波数 |
| タイマ0 | ５、６ | 980Hz |
| タイマ１ | ９、１０ | 490Hz |
| タイマ２ | ３、１１ | 490Hz |

1. Arduino NanoはツールメニューからプロセッサとしてATmega328p(Old　bootloader)を指定すること。そうしないと書き込めない（2018年11月26日追記）

# 接続

1. 車体インターフェイス表記:Pnn 例:P10はSHDN0の意味
2. A-D9等はArduinoのディジタル9番ピンの意味。E-xxx表記はESP32のピンの意味
3. VL53L0X(レーザー測距モジュール)
4. 1モジュール当たりに電源２本、I2C ２本とアドレス設定用１本（SHDN）が必要、つまり1モジュール当たり5本、ただし電源と信号線2本は共通で良い
5. 3モジュールの場合は電源2本、I2C 2本、SHDNモジュール数分(3本)の7本必要
6. 動作電圧は3.3Vから5.0V、自動車本体は4.5V供給可能であるが信号線最大電圧に合わせる必要があるのでP7(Arduino　V5、ESP３２　3.3V)から電源供給することにする
7. UART(シリアル通信)
8. Arduino

ハードウエアシリアルは１個しかない。通常その１個はPCでのシリアルモニタに使う。そのため他のモジュール(例えばVR3等)とシリアル通信をする場合はソフトシリアルを使う。シリアル通信用ピンは空いているピンをソフトウエア的に設定可能。例えばA6、A7やD7、D8などが使える。もちろんシリアルモニタを使わない、PCと接続しない使い方であればシリアル0が使える。

1. ESP32

ハードウエアシリアルは３個ある。シリアル0はPCと接続、シリアルモニタで使う。シリアル1は内部使用のため使えない。従ってシリアル2を使う。ピンはGPIO16と17を使う。

表 　ESP32のUART

|  |  |  |  |
| --- | --- | --- | --- |
| UART | RX | TX | 備考 |
| UART0 | GPIO3 | GPIO1 | USB書き込みで使用、使用推奨しない |
| UART1 | GPIO9 | GPIO10 | 通常使用不可 |
| UART2 | GPIO16 | GPIO17 | 今回採用 |

1. 電源の考え方

　ArduinoはUSB経由での電源供給は5Vになる。仕様では７Vから12VをVINに供給することになっている。ArduinoはVINの電源から内蔵レギュレータにより自分で内部電源用5Vを生成している。現在Arduino系CMの電源には昇圧型可変電圧型DC-DCコンバータ(MT3608等)を使っている。可変電圧型DC-DCコンバータで正確に3.3Vや5Vを生成しにくいため仕様に従ってVINに7.2V程度を供給している。DRV8835やVL53L0Xの電源供給はArduinoの供給能に頼っている。Arduinoは1ピン当たり最大40mAで全体で100mA供給できる。ESP32は供給可能な電力がもっと少ないのでESP32などから電源供給させずに3.3V固定型DC-DCコンバータなどを使いESP32への電源供給とともにP7(DRV8835やVL53L0Xへの電源供給ピン)へも並列で電源供給すると良いだろう。電圧固定型DC-DCコンバータであれば例えば5V供給するのであればArduinoもVINではなく5Vラインに直接電源供給できる。

1. I2C通信

　通常I2C通信の機器は機器毎に固有のアドレスを持たせてそのアドレスに対して通信する。逆に言うと複数の機器を同時に使う場合、それらの機器のアドレスは重複せずに一意である必要がある。今回VL53L0Xを複数使うがVL53L0Xは工場出荷時に全てに同じアドレスが設定されている。複数使うときにはアドレスの再設定が必要となる。VL53L0Xの仕様としてSHDN(シャットダウン)後にアドレスを送るとその機器のアドレスがその送ったアドレスに書き換わる機能がある。今回その機能を使って3個のVL53L0Xを使い分けることにした。書き換えられた新アドレスは機器に記憶されないのでVL53L0X起動時に常にCMから各VL53L0Xのアドレスを書き換えて使用するロジックを組む必要がある。

1. CMモジュール(Arduino等側)と車体側の接続
2. EX-IF（車体インターフェイス(車体側、図の上方が進行方向(前方))）

表 　第三世代　車体-CM接続インターフェイス(EX-IF)(旧表記)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| I/O | 用途 | ピン番号 | | 用途 | I/O |
| O | 電源(4.5V) | 1 | 8 | SDA:E-GPIO21 | I |
| O | GND | 2 | 9 | SCL:E-GPIO22 | I |
| I | AIN1:右折側:A3; E-GPIO04 | 3 | 10 | SHDN0(左):E-GPIO19 | I |
| I | AIN2:左折側:A11: E-GPIO26 | 4 | 11 | SHDN1(中):E-GPIO18 | I |
| I | BIN1:前進側:A5: E-GPIO27 | 5 | 12 | SHDN2(右):E-GPIO05 | I |
| I | BIN2:後進側:A6: E-GPIO25 | 6 | 13 | SERVO(操舵)用A9番ピン  E-GPIO32 | I |
| I | 信号最大電圧及び電力供給(通常コンピュータのVCC)(5Vまたは3.3V)VL53L0XとDRV8835のVCCへ | 7 | 14 | SERVO用A10番ピン  E-GPIO33 | I |
|  | 誤挿入防止用のダミーピン  (接続先は無し) | 7.5 |  |  |  |

P7はCM(ArduinoやESP32)の動作電圧と同じ電圧の「電源用」として車体側に供給すること。車体側ではP7を電源としてVL53L0XやDRV8835が動く。

P7はCM側がArduino系の場合は5Vラインに接続する。CM側がESP32の場合はDCDCコンバータの3.3V電源に直接接続する(ESP32の3.3Vラインには接続しない)

~~表 3車体-CM接続インターフェイス(縦表記)~~

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ~~ピン番号~~ | ~~I/O~~ | ~~用途~~ | ~~CM側接続先(ピン番号等)~~ | |
| ~~Arduino(A/D)~~ | ~~ESP32(GPIOピン番号)~~ |
| ~~1~~ | ~~O~~ | ~~電源(4.5V)~~ | ~~DCDCコンバータの入力~~ | |
| ~~2~~ | ~~O~~ | ~~GND~~ | | |
| ~~3~~ | ~~I~~ | ~~AIN1:右折側~~ | ~~3(D3)~~ | ~~04~~ |
| ~~4~~ | ~~I~~ | ~~AIN2:左折側~~ | ~~11(D11)~~ | ~~26~~ |
| ~~5~~ | ~~I~~ | ~~BIN1:前進側~~ | ~~5(D5)~~ | ~~27~~ |
| ~~6~~ | ~~I~~ | ~~BIN2:後進側~~ | ~~6(D6)~~ | ~~25~~ |
| ~~7~~ | ~~I~~ | ~~VL53L0XとDRV8835のVCCへ~~ | ~~5V出力ライン~~ | ~~(ESP32ではなく)DCDCコンバータの3.3V出力へ~~ |
| ~~7.5~~ |  | ~~誤挿入防止用のダミーピン~~  ~~(接続先は無し)~~ |  | |
| ~~8~~ | ~~I~~ | ~~SDA~~ | ~~18(A4)~~ | ~~21~~ |
| ~~9~~ | ~~I~~ | ~~SCL~~ | ~~19(A5)~~ | ~~22~~ |
| ~~10~~ | ~~I~~ | ~~SHDN0(左)~~ | ~~14(A0)~~ | ~~19~~ |
| ~~11~~ | ~~I~~ | ~~SHDN1(中)~~ | ~~15(A1)~~ | ~~18~~ |
| ~~12~~ | ~~I~~ | ~~SHDN2(右)~~ | ~~16(A2)~~ | ~~05~~ |
| ~~13~~ | ~~I~~ | ~~SERVO用~~ | ~~9(D9)~~ | ~~32~~ |
| ~~14~~ | ~~I~~ | ~~SERVO~~ | ~~10(D10)~~ | ~~33~~ |

表４は表１２に変更。表１２を参照。

I/Oの表記は車体側から見た表記。例えばP1はO、つまり出力になっている。これは車体側の電源4.5VをCM(コンピュータモジュール)側に供給するから車体側から見れば出力になる。

第三世代は赤外線測距モジュール利用は想定せずレーザー測距モジュールを想定する。

なおP7は今回「I」に変更した。理由としてモータドライバの信号線入力（AIN1など）が最大値でVCCとなっているため。モータドライバに直接乾電池からの電源供給すると4.5Vになる。モータドライバ自体は問題なく動くが信号入力電圧上限が4.5VとなってしまいArduinoの信号出力だと5Vが掛かってしまいオーバーする。ESP32だと3.3Vであるので上限自体はオーバーしないが半端な電圧となる。従ってモータドライバのVCCとCMのVCC（Arduino(5V)やESP32(3.3V)）を合わせるためCM側のコンピュータのVCCと同等の基本電圧を供給することとする。

~~8号車のP1は9V~~

1. DVR8835(モータドライバ)

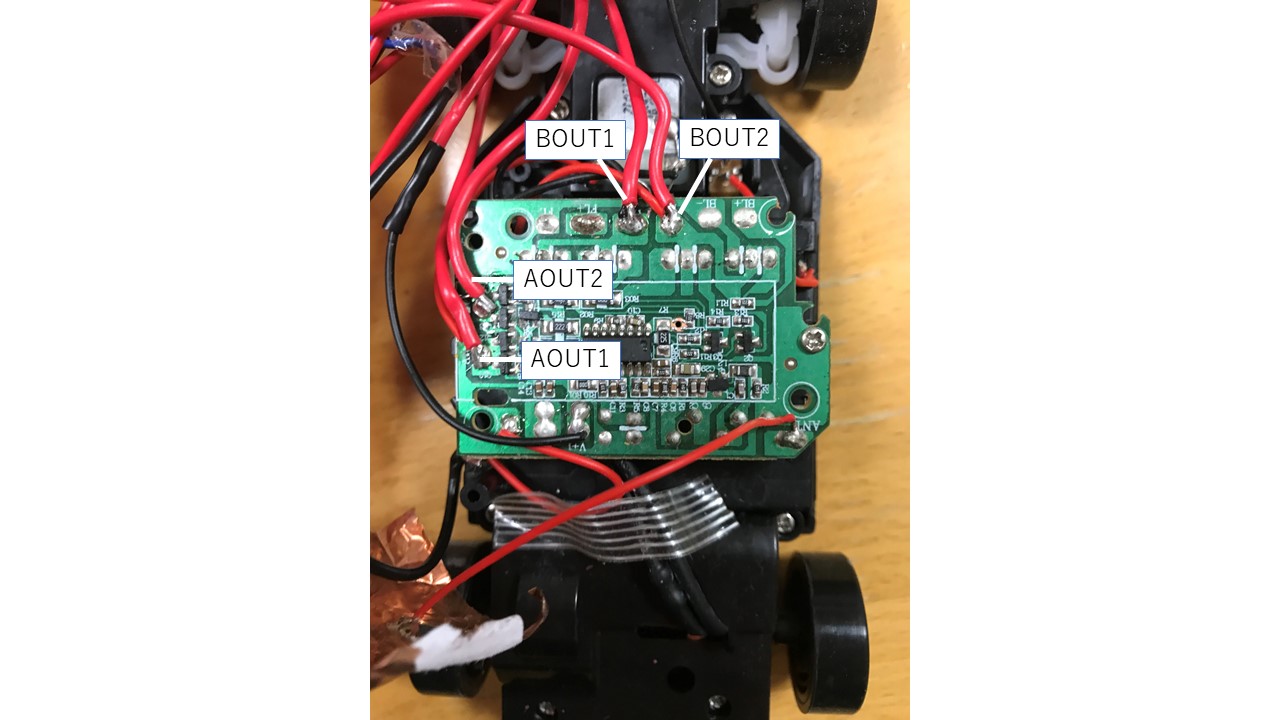


図 　24型結線図

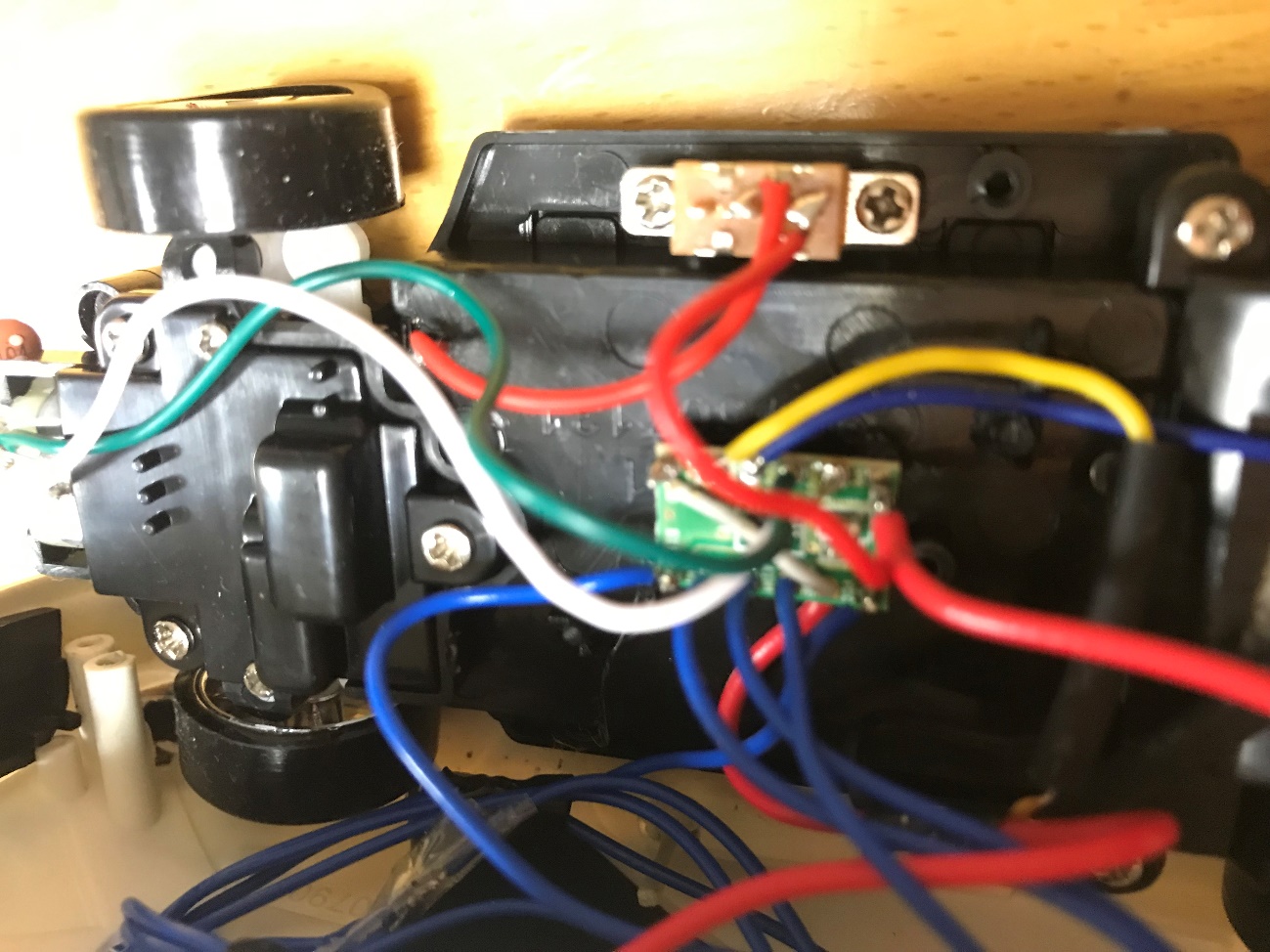


図 　32型結線図

24型(型番24-1-xx)はラジコン受信機を残しスイッチでラジコンとコンピュータコントロールを切り替えられるようにした。32型(型番 32-1-xx)は配線取り回しが厳しいのでラジコン受信機は撤去しコンピュータコントロールのみとした。

表 　32型DRV8835結線図

|  |  |  |  |
| --- | --- | --- | --- |
| 接続先 | 端子(左/右) | | 接続先 |
| 本体電源(スイッチ経由) | VM | VCC | P7 |
| 操舵モータ白線 | AOUT1 | MODE | グランドへ |
| 操舵モータ緑線 | AOUT2 | AIN1 | P3 |
| 走行モータ青線 | BOUT1 | AIN2 | P4 |
| 走行モータ黄線 | BOUT2 | BIN1 | P5 |
| グランド(乾電池マイナス側へ) | GND | BIN2 | P6 |

1. 型番32-1-xxタイプの第二パラメータが「1」のモデルはサーボモータを内蔵していないのでP13とP14は接続なし
2. 電源は本体スイッチ経由の「＋」電源を指す。

接続先

* + - 1. 本体スイッチ経由の「+」電源
      2. DRV8835のVM
      3. 車体-CMのP1

1. グランドは本体電池端子の「-」側に接続する

接続先

* + - 1. 本単電池端子「-」
      2. DRV8835のグランド
      3. DRV8835のMODEへ
      4. 車体-CMのP2
      5. 各VL53L0XのGND

1. VL53L0Xの結線
2. VIN、GND、SCL、SDAは３個の素子をそれぞれまとめて１本にして良い。例えば各VL53L0XのSCL３本をまとめて１本として、P9に接続する。個別に接続する必要があるのはシャットダウンライン（モジュールにより表記がことなるがSHDNやXSHUT等の表記になる）のみである。GPIOは使わない(結線せず)

接続先

* + - 1. 3本まとめたVINはP7へ
      2. 3本まとめたGNDはグランドへ(P2や乾電池端子「-」)
      3. 3本まとめたSDAはP8へ
      4. 3本まとめたSCLはP９へ
      5. 左VL53L0XのSHDNはP10へ
      6. 真ん中のVL53L0XのSHDNはP11へ
      7. 右のVL53L0XのSHDNはP12へ

1. BMX055の結線
2. JP6をショート(５V電源)
3. JP4とJP5をショート（I2C内部プルアップ有効）

第三世代はモータドライバは本体に搭載することにした。理由はコンピュータのVCCと動作電圧を合わせた方が良いからである。

またDRV８８３５は超低消費電力（通常時450μA、最大2000μA（2mA））となっておりArduinoの5V端子（40mA）でも十分動作可能である。

動作電圧は７から12Vで信号電圧は5V。

ESP32は動作電圧3.3V（VCC）と信号電圧が同じ。動作電圧と信号電圧が同じ場合はDRV8835のVCCはコンピュータと同じ電源から取ればよい。問題になるのはコンピュータの動作電圧と信号電圧が異なる場合、この場合はコンピュータ（例えばArduino）より5Vを電源供給する。

また測距モジュールも同じSPRESENSEはI2C信号が1.8VなのでI2Cコンバータで1.8Vと5Vを接続することにする。

表 　Arduino nano(USB端子を下とした図)

|  |  |  |  |
| --- | --- | --- | --- |
| 接続先 | Arduinoピン | | 接続先 |
|  | TX1 | VIN | 電源(DC-DCコンバータからの出力、7.5V) |
|  | RX0 | GND | グランド |
|  | RST | RST |  |
| P2 | GND | 5V | P7 |
|  | D2 | A7 |  |
| P3(AIN1) | D3[3] | A6 |  |
|  | D4 | A5[19] | P9(SCL) |
| P5(BIN1) | D5[5] | A4[18] | P8(SDA) |
| P6(BIN2) | D6[6] | A3 |  |
|  | D7 | A2[16] | P12(SHDN2) |
|  | D8 | A1[15] | P11(SHDN1) |
| P13(サーボ) | D9[9] | A0[14] | P10(SHDN0) |
| P14(サーボ) | D10[10] | REF |  |
| P4(AIN2) | D11[11] | 3V3 |  |
|  | D12 | D13 |  |

Anはアナログピン、Dnはディジタルピン、[n]は物理ピン

通常はVINに7.2V程度を供給しArduinoの内蔵レギュレータ経由でArduino自身に5V電圧を生成させる構成とする(標準構成)。VL53L0XやDRV8835はArduinoやESP32の生成された5Vや3.3Vの供給可能電力で動ける。ただしこれらCM上に他のこれらより消費電力が多そうなモジュールを搭載する場合（例えばVR3）はArduinoやESP32の生成電力では足りない可能性があるのでその場合はArduinoもVINではなく直接5Vや3.3VにDC-DCコンバータ経由で供給する構成にする。DC-DCコンバータより並列でArduino等やモジュールに電源供給する。理由として7.2Vと５V(や3.3V)の2系統の電源を用意するのは回路が複雑になるから避けるためである。特にESP32では直接3,3Vを供給した方が手っ取り早い。

1. ESP32（Devkit）のPWM関係情報

ESP32自体はPWM用に16チャンネルある。

そのうちDevkitなどが内部で使用済で使えないピンがある。

表 　ESP32/ESP32DevKit　PWM関係ピンアサイン

|  |  |  |  |
| --- | --- | --- | --- |
| ESP32 | | ESP32Devkit | 備考・説明 |
| アナログピン番号 | GPIO番号 | 使用不可理由 |  |
| ***A0*** | ***36(使用不可)*** |  |  |
| ***A3*** | ***39(使用不可)*** |  |  |
| A4 | 32 |  | サーボで使用(9番相当) |
| A5 | 33 |  | サーボで使用(10番相当) |
| ***A6*** | ***34(使用不可)*** |  |  |
| ***A7*** | ***35(使用不可)*** |  |  |
| A10 | 4 |  | AIN1とする |
| ***A11*** | ***0*** | ***ブート時使用*** |  |
| ***A12*** | ***2*** | ***ブート時使用*** |  |
| ***A13*** | ***15*** | ***U0TXDデバッグ用*** |  |
| ***A14*** | ***13*** | ***使用不可(秋月)*** |  |
| ***A15*** | ***12*** | ***LDO電圧設定用*** |  |
| A16 | 14 |  |  |
| A17 | 27 |  | BIN1とする |
| A18 | 25 |  | BIN2とする |
| A19 | 26 |  | AIN2とする |

表 　ESP32DevKit接続図(USB端子を下とした図)

|  |  |  |  |
| --- | --- | --- | --- |
| 接続先 | ESP32PIN | | 接続先 |
| 電源入力 | 3.3V | GND | P2 |
|  | EN | GPIO23 |  |
| 出力には使えない | GPIO36 | GPIO22 | P9(SCL) |
| GPIO39 | GPIO01 | UART通信用  通常は使わないこと |
| GPIO34 | GPIO03 |
| GPIO35 | GPIO21 | P8(SDA) |
| P13(SERVO用) | GPIO32 | GND |  |
| P14(SERVO用) | GPIO33 | GPIO19 | P10(SHDN0) |
| P6(BIN2) | GPIO25 | GPIO18 | P11(SHDN1) |
| P4(AIN2) | GPIO26 | GPIO05 | P12(SHDN2) |
| P5(BIN1) | GPIO27 | GPIO17 | UART2TX(VR3へ) |
|  | GPIO14 | GPIO16 | UART2RX(VR3へ) |
| LDO電圧設定用 | ***GPIO12*** | GPIO04 | P3(AIN1) |
|  | GND | ***GPIO00*** | ブート時使用される |
| 使用不可(秋月Webより) | ***GPIO13*** | ***GPIO02*** |
|  | ***GPIO09*** | ***GPIO15*** | U0TXDデバッグ用 |
|  | ***GPIO10*** | ***GPIO08*** |  |
|  | ***GPIO11*** | ***GPIO07*** |  |
|  | 5V | ***GPIO06*** |  |

*斜体ゴシック*のピンは内部で使用されているため利用禁止

（参考：過去情報）ESP32電源用に三端子レギュレータシャープ製PQ3RD23を採用。３、４号車の三端子レギュレータは東芝製TA48033Sを使用していた。これはArduinoの電源ではなくシャープ製赤外線測距モジュールGP2Y0E02Aが3.3V動作であったため採用していたがTA48033Sは最大1Aまでしか対応できない。そこでより容量の大きい三端子レギュレータを探して採用した。PQ3RD23は最大２Aまで対応可能。ESP32は最大消費電力1.5A程度とのことで容量的には範囲内。ちなみに１、２号車のシャープ製赤外線測距モジュールの電源電圧は5V。

レーザー測距モジュールVL53L0Xの電源は3.3Vから5.0V、ただしDRV8835は乾電池から電源は取れるものの信号線の最大入力がVCCとなっているのでCM側のVCCと電圧を合わせる必要があるからP7から電源を取る。

なお現在は三端子レギュレータではなくDC-DCコンバータで電源生成。

第二世代ではモータドライバはCM側に搭載していた。

IN-IFは廃止

9号車のスイッチ部分。

進行方向左がオン（左と中が導通）

左側に電源からの9Vが供給されている。

オンの時、真ん中下側に9V供給される。

1. ラジコン切替スイッチ

切り替えスイッチは、下から見て左がCM、右がMC

センサは左が０、真ん中が１、右が２とする。

DRVのVMへ

P1へ

表 　VL530LXとDRV8835仕様

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | GY-VL530XV2  (中国製) | VL53L0X  (ストロベリーリナックス製) | DRV8835 |  |
| 電源 | VIN:2.6-5.5V | VDD:3.3-5.0v | Vcc:2-7V |  |
| 信号線電圧 | SDA/SDC  Low=0,HIGH=VIN | 同左 | Vin:最大がVcc |  |
| VM |  |  | 0-7V |  |
| その他 | VDDは直接の電源、通常はVCCを使うこと |  |  |  |
| 消費電流 | 10mA(最大40mA) | 同左 | 2mA |  |
| 測定不能時値 | 8190 | 0 |  |  |

表 　CM用コンピュータ仕様

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Arduino Nano | ESP32 | RasPiZero | PasPi PICO | Obniz |
| 電源 | 7-12V  VINへ | 2.2-3.6V(3V3へ入力の場合)  3.7-6.0V(VINへ入力の場合) | 5V(2番ピンまたは4番ピン) | 39番ピン(VSYS)  1.8-5.5V | ５V  基板裏側 |
| 内部動作 | 5V | 3.3V | 3.3V |  |  |
| I/Oピン電圧 | 5V | 3.3V | 3.3V | 3.3V | 5V(電圧選択可能) |
| I/Oピン電力供給能 | ~~40mA~~  ~~全体で100mA~~  800mA | 最大12mA | 16mA  全体で50mA | 300mA以下 | 5Vモードで1A |
| 消費電力(電流) | 65mA | WiFi時160-260mA | 125mA  WiFi時270mA | 5V給電時95mA程度 | 平均170mA(WiFi不使用時) |
| USBコネクタ | ミニB | マイクロB | マイクロB | マイクロB |  |
| PWM | 6本 | 4本 | ハード2本 | 16本 | ６本 |
| 推奨電圧変換 | MT3608系 | PQ3RD23 | MT3608系  またはTPS63020 | ~~PQ3RD23またはTPS63020~~  入力電圧が1.8V-5.5Vなので車体側電圧が3Vや4.5VであればDCDC不要で直接供給する | MT3608系  またはTPS63020 | |
| I/O用電圧 |  |  | PQ3RD23  またはSOT-23AP7333 | （不要） |  |

Nanoの5V出力800mA(0.8A)については下記の公式ページ参照

https://docs.arduino.cc/learn/electronics/power-pins/

|  |  |  |  |
| --- | --- | --- | --- |
|  | Spresense |  |  |
| 電源 | 5V BtoBの2番と4番ピンへ |  |  |
| 内部動作 |  |  |  |
| I/Oピン電圧 | 1.8V |  |  |
| I/Oピン電力供給能 | 6mA |  |  |
| 消費電力 | 500mA |  |  |
| I2C電圧 | 1.8V |  |  |
| USBコネクタ | マイクロB |  |  |
|  |  |  |  |

表 　DC-DCコンバータ仕様

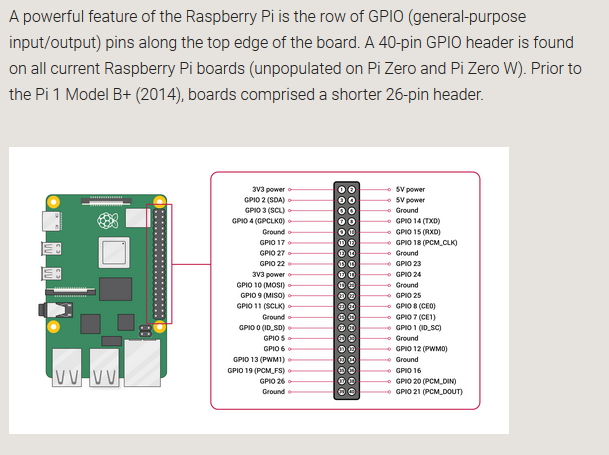
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| コンバータ | XL6009使用 | LM2577使用 | LM2596 | MT3608 | PQ3RD23  (三端子レギュレータ) |
| 昇降圧タイプ | 昇降圧 | 昇降圧 | 降圧 | 昇圧 | 降圧 |
| 入力電圧(V) | 3.8-32 | 3.3-35 | 4-35 | 2-24 | ～20V |
| 出力電圧(V) | 1.25-35 | 1.25-30 | 1.23-30 | 最大28 | 3.3V |
| 出力など | 入力最高3A | 最大2A | 2A(ヒートシンク時3A) | 最大2A出力 | 最大出力2A |
| 入出力絶縁 | 非絶縁 | 不明 | 非絶縁 | 不明 |  |
| 備考 |  |  |  |  | 入出力電圧差0.5Vで良い |

|  |  |  |  |
| --- | --- | --- | --- |
| コンバータ | TPS63000 | TPS63070 | TPS63020 |
| 昇降圧タイプ | 昇降圧 | 昇降圧 | 昇降圧 |
| 入力電圧(V) | 1.8-5.5 | 2-16 | 1.8-5.5 |
| 出力電圧(V) | 3.3か5の切り替え | 2.5-9 | 3.3か5の切り替え |
| 出力など | 入力:1.8A  出力:3.3V降圧時1.2A | 昇降圧とも2A | 昇圧２A  降圧3A |
| 入出力絶縁 | 不明 | 不明 |  |

# CM接続

1. CM
2. CM3-AN系
3. DC-DCコンバータの入力と出力のGNDを接続する
4. DC-DCコンバータの入力側に4.5V直流電源(乾電池直列3本等)を接続し出力側の電圧をテスタ等で計測する
5. 可変ボリュームを回して出力が7.2V程度に調整する
6. ホットメルトグルー等でボリュームを固定
7. これらを台数分作成
8. 左側用8ピンソケットと右側用7ピンソケットを用意。8ピンヘッダと7ピンヘッダを刺す。その状態でD型基盤をはんだ付け開始する
9. モジュールの基盤への取り付けは通常のピンヘッダで良い。細ピンヘッダを使う必要はない
10. 表2及び表4を参考にして配線する。基本はP1からの電源をDC-DCコンバータの入力に接続し、DC-DCコンバータの出力をArduinoのVINに接続する。その後は表3に従い結線する。グランドは各モジュール全部共通接続
11. CM3-AN系動作確認方法
12. 車体に電池を入れたものを用意する。車体側のスイッチは「この時点ではOFFとすること」
13. 車体にCMを接続しUSBケーブルでPCに接続する
14. PC側の接続COMポート番号をデバイスマネージャーで確認する。接続されるCOMポート番号は都度変わるので接続都度確認した方が良い
15. Arduino IDEを立ち上げる。
    * + - 1. ボードはArduino nano
          2. ブートローダは「同oldブートローダ」
          3. ポート設定はデバイスマネージャーで確認したCMモジュールが接続されているポートに設定
16. 各接続チェック用プログラム「CarbodyhealthCheckxxxxxxxx」をロード。「xxxxxxxx」は日付表示であり、基本的には最新の日付のものを使うこと
17. このプログラムはVL530LXとのI2C通信や操舵・走行関係の結線が正しく行われているか確認するためのものである
18. Arduino IDEのシリアルコンソールを開いておく
19. プログラムをコンパイルしCMへロードする
20. CMへのプログラムロードが正常終了すると自動的にプログラムが実行されシリアルコンソールへ各３個のVL53L0Xの測距値がそれぞれ表示されるので各センサ(VL530LX)に対して測距対象（例えば手など）を動かしてシリアルモニタ上に表示される測定結果が正しいか目視で確認する。これによりVL530LXに関連するI2C接続の正確性の確認ができる。VL53L0XへはArduinoから電源供給されるので車体の本体スイッチがOFFでもArduinoへ電源供給(例えばPCから)されていればVL53L0Xは稼働できるのでこのような事が可能になる
21. 次にPCと接続しているUSBケーブルを抜く。CMへの電源供給がストップしCMは停止する
22. ここで「OFFになっている車体側スイッチをONにする」
23. 車体側からCMへの電源供給が開始されるのでCMが再び起動する
24. プログラムは、左右のそれぞれの操舵と短く走行用車輪をそれぞれ前後に回転させる。これら動作が確認できたら操舵・走行関係の接続の正確性の確認ができたといえる
25. Arduino　IDEのクローズや本体電源スイッチをOFFにし、終了
26. RumiCar32-1系製作方法
27. 本体を分解し、ラジコン受信機とアンテナを撤去
28. 残す配線や必要な端子は、各モータの線(合計4本)と本体電源スイッチ経由の「+」電源、グランド用の乾電池端子「-」のみ
29. VL53L0Xの配線用にボンネットにドリルで穴を開ける。ドリルで開け、リーマで広げ、穴の縁をカッター等でなめバリを取る

https://www.raspberrypi.org/documentation/usage/gpio/



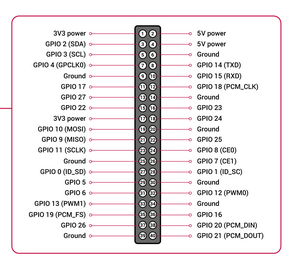


図 RumiCar インターフェイス

表 　CM接続インターフェイス

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | I/O | 用途 | | CM側接続先(ピン番号等) | | Raspberry Pi Zero W | Pico | Obniz | SPRESENSE |
| Arduino(A/D) | ESP32(GPIOピン番号) | RasPi ピン番号(GPIO番号) | |  |  |
| 1 | O | 電源(4.5V) | | DCDCコンバータの入力 | | | | | |
| 5V | 3.3V | 5V | 3.3V(1.8から5.5) | 5V | |
| 2 | O |  | GND | | | | | | |
| 3 | I | AIN1:右折側 | | 3(D3) | 04 | 11(17) | 27(21) | io0 | 39(D06) |
| 4 | I | AIN2:左折側 | | 11(D11) | 26 | 13(27) | 29(22) | io1 | 41(D05) |
| 5 | I | BIN1:前進側 | | 5(D5) | 27 | 32(12)PWM0 | 31(26) | io2 | 31(D09) |
| 6 | I | BIN2:後進側 | | 6(D6) | 25 | 33(13)PWM1 | 32(27) | io3 | 45(D03) |
| 7 | I | VL53L0XとDRV8835のVCCへ | | 5V出力ライン | (ESP32ではなく)DCDCコンバータの3.3V出力へ | 3.3V出力のDCDCコンバータの出力へ | 36(3V3) | DCDCコンバータの出力(5V)へ | |
| 7.5 |  | 誤挿入防止用のダミーピン  (接続先は無し) | | | | | | | |
| 8 | I | SDA | | 18(A4) | 21 | 3(2) | 16(12) | io4 | 21(D14) |
| 9 | I | SCL | | 19(A5) | 22 | 5(3) | 17(13) | io5 | 19(D15) |
| 10 | I | SHDN0(左) | | 14(A0) | 19 | 16(23) | 11(8) | io6 | 28(D02) |
| 11 | I | SHDN1(中) | | 15(A1) | 18 | 18(24) | 10(7) | io7 | 50(D07) |
| 12 | I | SHDN2(右) | | 16(A2) | 05 | 22(25) | 9(6) | io8 | 60(D41) |
| 13 | I | SERVO(操舵)用 | | 9(D9) | 32 | 15(22) | 7(5) | io9 | D08 |
| 14 | I | SERVO用 | | 10(D10) | 33 | 37(26) | 6(4) | io10 | D04 |
| ボードの電源 |  |  | |  | 2 |  | 39(VSYS) |  |  |

表 　I2Cアドレス

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| センサ | | | アドレス | 設定方法 |
| VL53L0X  (レーザ測距モジュール) | Sensor0 | 左 | 0x20 | ソフトウエアにより起動時に変更処理実行 |
| Senseo1 | 中央 | 0x21 |
| Sensor2 | 右 | 0x22 |
| Sensor3 | 後部 | 0x52 | 固定値(初期値) |
| BMX０５５  (9軸センサ) | 加速度 | | 0x19 | 基板上でジャンパ設定 |
| ジャイロ | | 0x69 |
| 磁気 | | 0x13 |