

Ditel モータードライバ

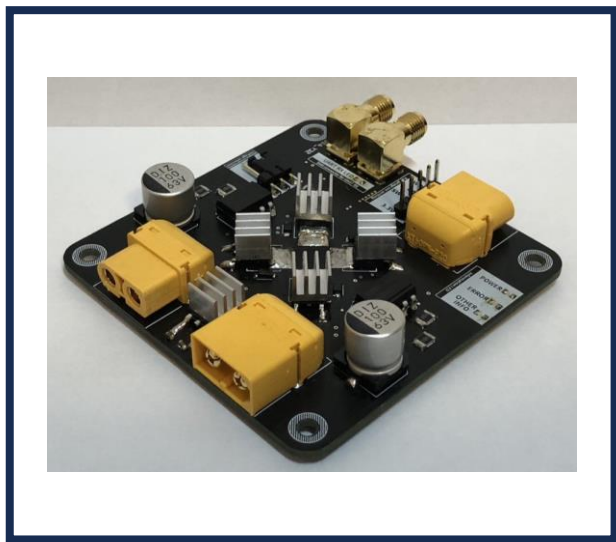
D-M-003

D-M-003(以降本モタドラ)は UART 通信を用いて操作できる DC モータ駆動用フルブリッジドライバモジュールです。

7~30V までの広い範囲の電圧をサポートし、6A までの電流を制御することができます。

また、正転/逆転/ブレーキ/ニュートラルの 4 つのモードを利用することができます。

さらに、タイムアウト機能を搭載しているため、思わぬ事故を防ぐことができ、安全性も高いです。



特徴

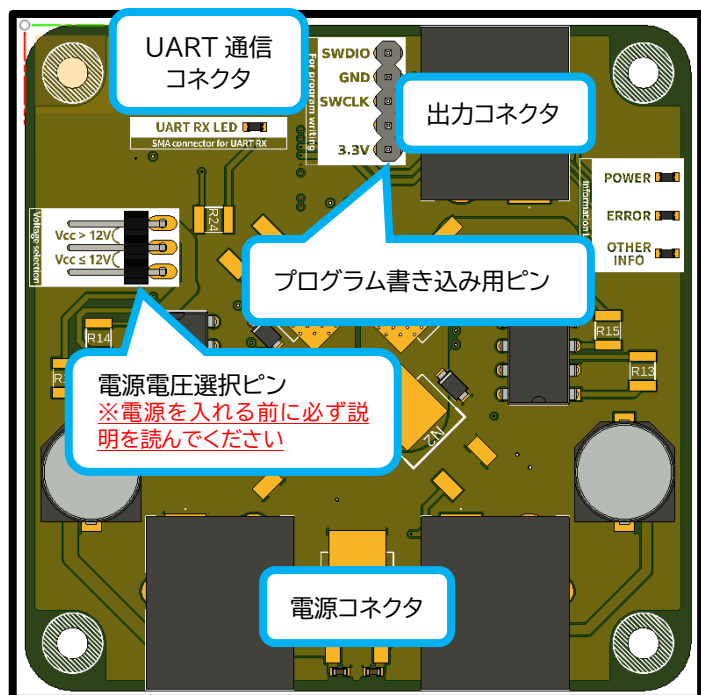
- ・電源電圧 : 30V(最大)
- ・出力電流 : 6A(最大)
- ・UART 通信による操作
- ・正転/逆転/ブレーキ/ニュートラル
- ・タイムアウト機能搭載

コネクタ・ピン説明

各コネクタ・ピンの役割は右の図の通りです。

また、本モタドラでは UART 通信コネクタとして SMA コネクタを使用しています。そのため通信には同軸ケーブルを使用しますが、内部導体はマイコンからの通信(マイコンの TX ピン)と、外部導体は GND と接続してください。

また、右の図にもある通り電源選択ピンは接続を間違えると本モタドラが故障する可能性があります。必ず使用する前に本データシート内の”電源を入れる前に”という項目を読んでください。



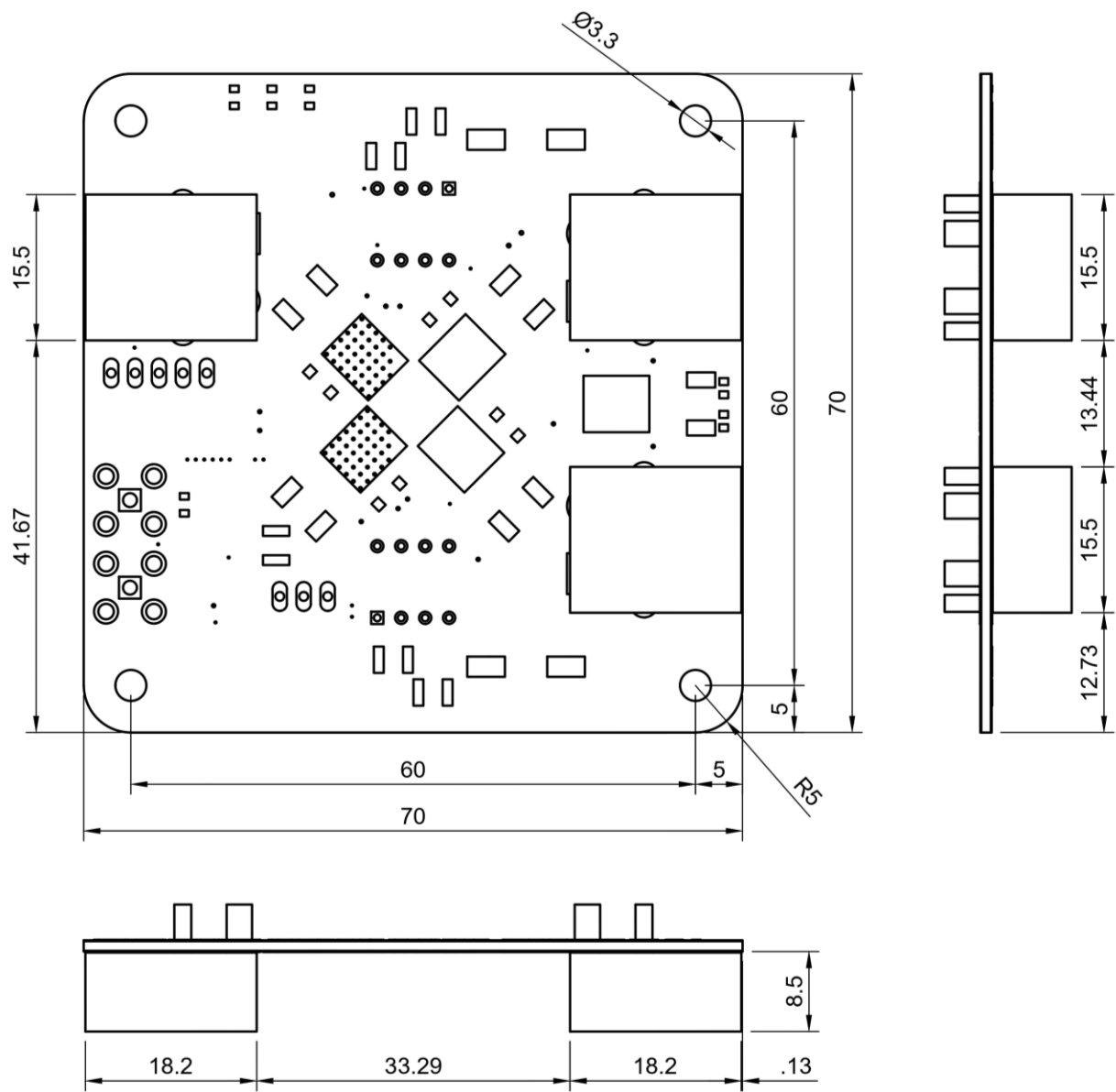
絶対最大定格

項目	記号	定格	単位
電源電圧	V_{CC}	0~35	V
入力電圧	V_{IN}	0~35	V
入力電流	I_{IN}	0~6	A
信号電圧	V_{SIG}	-0.3~4.0	V

動作範囲

項目	記号	定格	単位
電源電圧	V_{CC}	7~30	V
入力電圧	V_{IN}	7~30	V
入力電流	I_{IN}	0~6	A
信号電圧	V_{SIG}	3.3	V

外形図



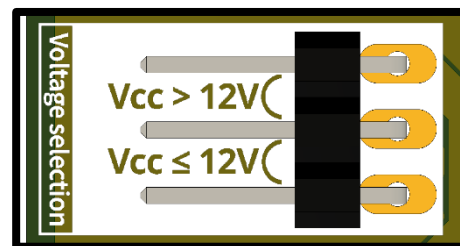
使用方法

電源を入れる前に(必ず読んでください)

本モータードライバは電源を投入する前に“Voltage selection”(右の画像)のピン接続を適切に行う必要があります。

電源電圧が 12V 以下である場合は真ん中と下のピンを短絡させて、12V より高い場合は真ん中と上のピンを短絡させてください。

※**真ん中と下のピンを短絡させた状態で 12V より高い電圧をかけないでください!!**



通信方法

本モータードライバは UART 通信でデータを送信することで、任意の回転数及び回転方向を指示することができます。通信の詳細は以下の通りです。

内容	詳細
通信速度	115200 Bits/s
データサイズ	4 bytes

または以下の内容を送信することで操作できます。

1 バイト目	2 バイト目	3 バイト目	4 バイト目
0xFE	“ドライバのアドレス”	“コマンド”	“その他の情報”

ドライバのアドレスはモータードライバ裏側に記載してあります。またコマンドについては下記の表にまとめた通りで、送信するコマンドによって“その他の情報”に記入する内容も異なってきます。

コマンド番号	コマンドの内容	“その他の情報”に記入する内容
0xA0	正転	“回転数(0~200)”
0xA1	逆転	“回転数(0~200)”
0xA2	ニュートラル	0xA5
0xA3	ブレーキ	0xA5
0xA4	タイムアウト機能の設定	“タイムアウトの時間 1~200…タイムアウトの時間(ms) 0…タイムアウト機能を OFF にする”

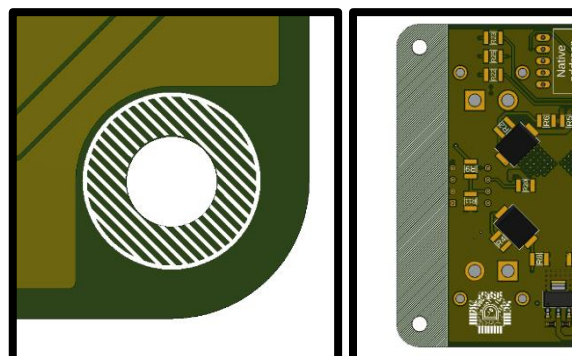
※タイムアウト機能の設定について任意の設定にするためには毎回起動時にコマンドを送信する必要があります。(タイムアウト機能は起動時に自動的に 50ms になります)

※回転させ続けるためには設定した時間(設定していない場合は 50ms)より短い間隔でデータを送信する必要があります。

※Arduino 言語のサンプルプログラムは付録 1 に記載してあります。

ロボットへの取り付け方法

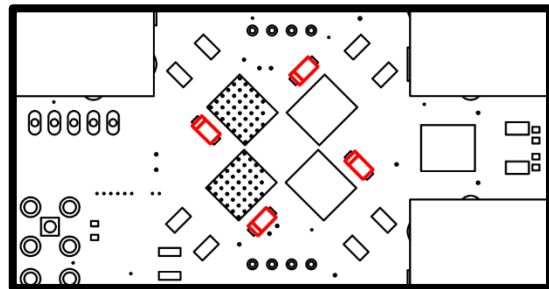
本モタドラはロボット等への取り付けに M3 のねじを使用することを想定して設計されています。外形図を参考にして必要な穴を取り付ける側に開けてください。また、加重は右の図のような斜線がある箇所以外に加えないでください。斜線がない部分は配線が通っているため、マスクがはがれ、本モタドラの寿命が縮まる可能性があります。



D-M-003 を作成するにあたって

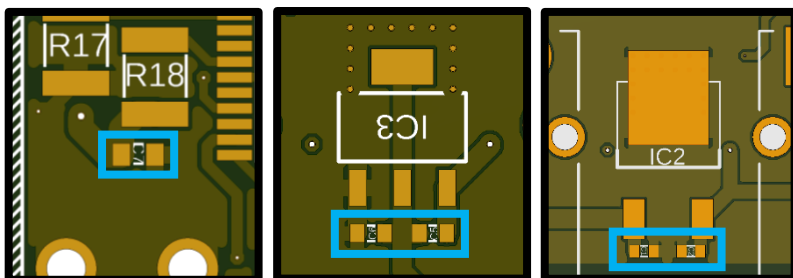
ダイオードの方向

D5, D6, D7, D8 について、シルク印刷の精度の都合で極性を基板に記載できていません。そのため右の写真と同様の向きになるようにはんだ付けをしてください。



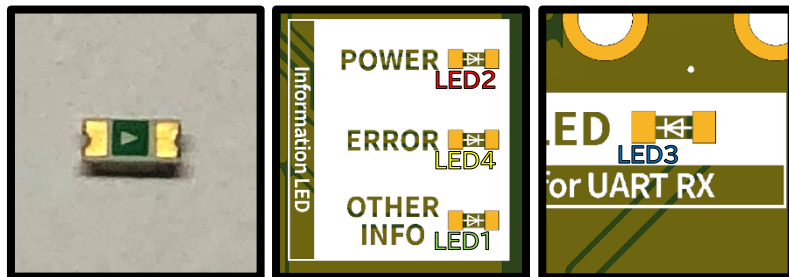
チップキャパシタの取り付け位置

C1, C2, C5, C6, C7 について、シルク印刷の精度の都合でパーツ名を基板に記載できていません。そのため右の写真において、水色の四角に囲まれているパッドに C1, C2, C5, C6, C7 をそれぞれはんだ付けしてください。



LED の取り付け位置

LED1, LED2, LED3, LED4 について、場所の都合でパーツ名を基板に記載できていません。そのため右の写真において、それぞれ該当するパーツ名の場所にはんだ付けしてください。また LED を取り付ける向きについては基板に印刷されている回路図記号の三角形が指す向きと、右の写真のように LED の裏側に印刷されている三角形が指す向きが一致するように取り付けてください。



プログラムの書き込み

本モトドラは STM32 マイコンチップを使用しているため、動作させるためにはプログラムを書き込む必要があります。

https://github.com/Ditel252/Ditel_MotorDriver_D-M-003

上記のサイトにアクセスし、Program_D_M_003.bin をダウンロードしたうえで ST-LINK 等を使用して本モトドラとパソコンを接続し、上記のファイルを書き込んでください。付録 2 として Nucleo F446RE を使用して書き込む際の接続の写真を載せているため参考にしてください。

部品表

写真	部品の種類	型番	パーツ	部品数
 ※ ¹	ピンヘッダ(オス)	PINHD-1X3/90	JP1.	1
 ※ ²	ピンヘッダ(オス)	PINHD-1X5	JP2.	1
 ※ ³	キャパシタ	63SGV100M10X10.5	C3 C4.	2
 ※ ⁴	三端子レギュレータ	AZ1117CH-3.3TRG1	IC3.	1
 ※ ⁵	キャパシタ	GRM188F11H104ZA01	C1 C2 C5 C6 C7.	5
 ※ ⁶	ダイオード	GS1010FL	D5 D6 D7 D8.	4
 ※ ⁷	三端子レギュレータ	NJM7812SDL1	IC2.	1
 ※ ⁸	MOSFET(Nch)	NP52N06SLG-E1-AY	N1 N2.	2
	抵抗	RK73B2ETTD101J	R1 R2 R3 R4 R13 R14 R15 R16 R22 R23 R24 R25.	12

	抵抗	RK73B2ETTD103J	R17 R18.	2
	抵抗	RK73B2ETTD132J	R9 R10 R11 R12.	4
	抵抗	RK73B2ETTD223J	R5 R6 R7 R8.	4
	抵抗	RK73B2ETTD681J	R20 R21.	2
 ※9	SMA コネクタ	S-037-TGG	J1 J2.	2
 ※10	ダイオード	SK54	D1 D2 D3 D4.	4
 ※11	LED(緑)	SML-E12P8WT86	LED1.	1
 ※11	LED(赤)	SML-E12V8WT86	LED2.	1
 ※11	LED(青)	SMLE13BC8TT86	LED3.	1
 ※11	LED(黄)	SML-E12Y8WT86	LED4.	1

 ※ ¹²	マイコンチップ	STM32L010F4P6	IC1.	1
 ※ ¹³	MOSFET(Pch)	TJ60S04M3L	P1 P2.	2
 ※ ¹⁴	フォトカプラ	TLP621-2XGB	IC4 IC5.	2
 ※ ¹⁵	XT60 コネクタ (メス)	XT60PW-F	PS1 PS2.	2
 ※ ¹⁶	XT60 コネクタ (オス)	XT60PW-M	PS3	1

※¹～※¹⁶…いずれの画像も秋月電子通商様のホームページより拝借したものです。具体的な URL については”出典”に記載しています。

付録 1

以下のサンプルプログラムにおいて想定されている構成は以下の通りです。

言語 : Arduino 言語

マイコン: Nucleo F401RE

<サンプルプログラム>

```
#include <Arduino.h>
#include <HardwareSerial.h>

#define PIN_UART_TX PA_2    //TX のピン
#define PIN_UART_RX PA_3    //RX のピン

#define ADDRESS 0x01        //モータードライバのアドレス

#define COMMAND_NORMAL      0xA0    //コマンド:正転
#define COMMAND_REVERSAL    0xA1    //コマンド:逆転
#define COMMAND_NEUTRAL     0xA2    //コマンド:ニュートラル
#define COMMAND_BREAK       0xA3    //コマンド:ブレーキ
#define COMMAND_SET_TIMEOUT 0xA4    //コマンド:タイムアウト機能の設定
```

```
HardwareSerial serialx(PIN_UART_RX, PIN_UART_TX);

void setup(){
    serialx.begin(115200);  //通信を開始

    delay(1000);    //1 秒待つ
}

void loop(){
    for(int i = 0; i <= 20; i++){    //正転の回転数をだんだんと上げていく
        serialx.write(0xFE);
        serialx.write(ADDRESS);
        serialx.write(COMMAND_NORMAL);
        serialx.write(i * 10);

        delay(40);
    }

    serialx.write(0xFE);    //ニュートラル
    serialx.write(ADDRESS);
    serialx.write(COMMAND_NEUTRAL);
    serialx.write(0xA5);

    delay(500);

    serialx.write(0xFE);    //ブレーキ
    serialx.write(ADDRESS);
    serialx.write(COMMAND_BREAK);
    serialx.write(0xA5);

    for(int i = 0; i <= 20; i++){    //逆転の回転数をだんだんと上げていく
        serialx.write(0xFE);
        serialx.write(ADDRESS);
        serialx.write(COMMAND_REVERSAL);
        serialx.write(i * 10);

        delay(40);
    }

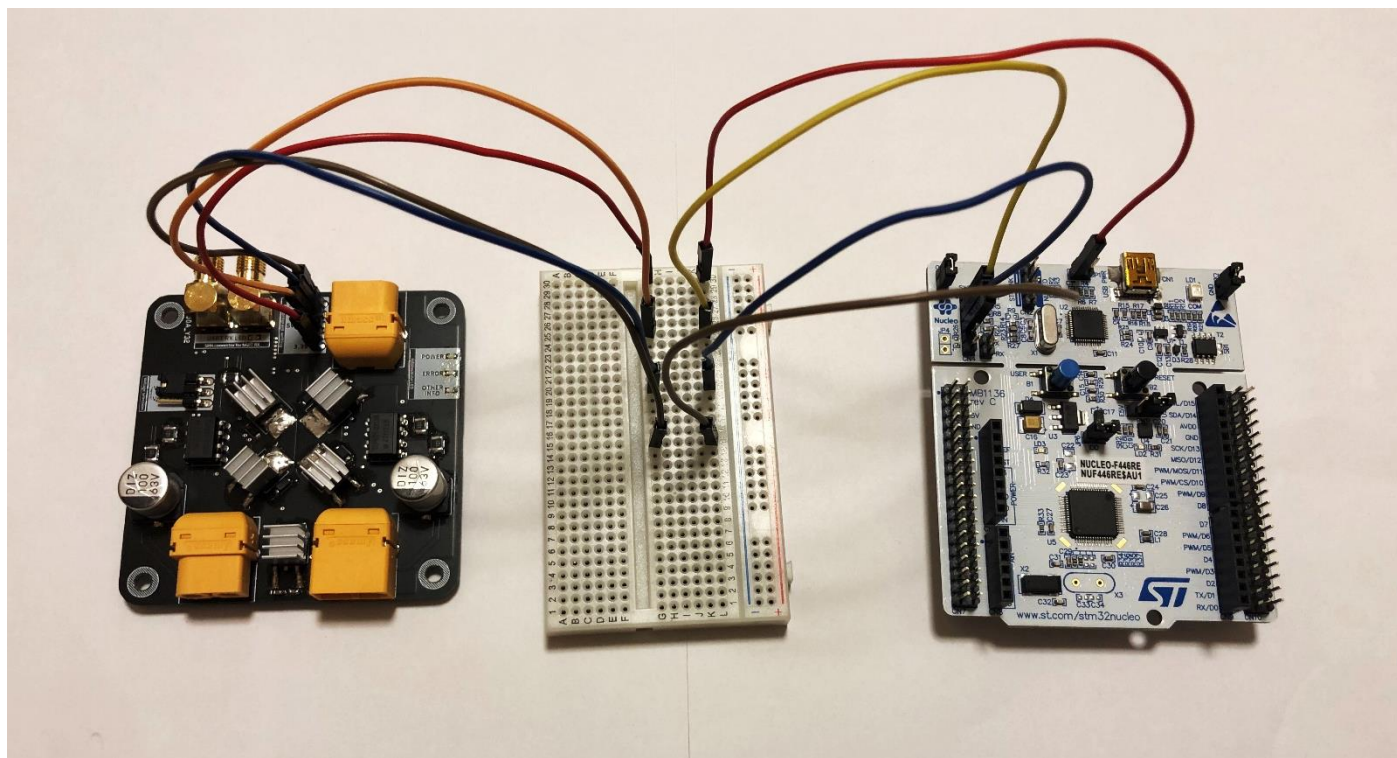
    delay(1000);

    serialx.write(0xFE);    //ブレーキ
```



```
serialx.write(ADDRESS);  
serialx.write(COMMAND_BREAK);  
serialx.write(0xA5);  
  
delay(100);  
}
```

付録 2



※ST-LINK ピンは必ず開放してください。

出典

- ※¹ <https://akizukidenshi.com/catalog/g/gC-01627/>
- ※² <https://akizukidenshi.com/catalog/g/gC-00167/>
- ※³ <https://akizukidenshi.com/catalog/g/gP-12940/>
- ※⁴ <https://akizukidenshi.com/catalog/g/gI-16155/>
- ※⁵ <https://akizukidenshi.com/catalog/g/gP-13374/>
- ※⁶ <https://akizukidenshi.com/catalog/g/gI-06014/>
- ※⁷ <https://akizukidenshi.com/catalog/g/gI-10978/>
- ※⁸ <https://akizukidenshi.com/catalog/g/gI-09928/>
- ※⁹ <https://akizukidenshi.com/catalog/g/gC-02569/>
- ※¹⁰ <https://akizukidenshi.com/catalog/g/gI-04128/>
- ※¹¹ <https://akizukidenshi.com/catalog/g/gI-11878/>
- ※¹² <https://akizukidenshi.com/catalog/g/gI-15689/>
- ※¹³ <https://akizukidenshi.com/catalog/g/gI-16297/>
- ※¹⁴ <https://akizukidenshi.com/catalog/g/gI-16465/>
- ※¹⁵ <https://akizukidenshi.com/catalog/g/gC-18155/>

※¹⁶ <https://akizukidenshi.com/catalog/g/gC-18156/>