

AIchipV3マニュアル

目次

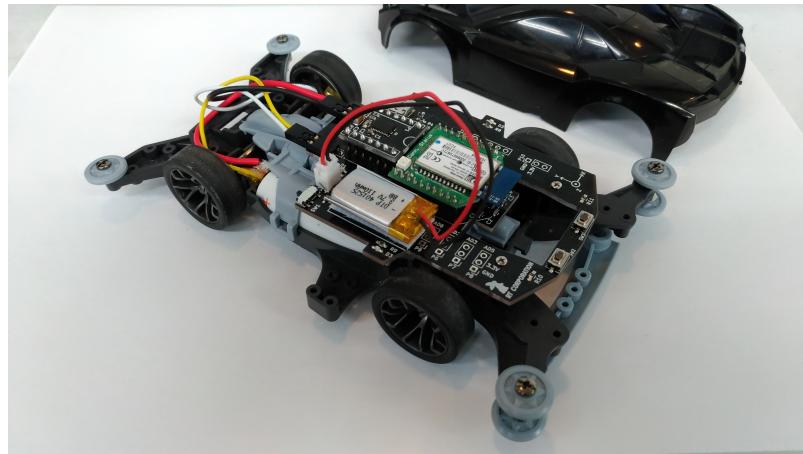
- [はじめに](#)
- [AIchipの使い方](#)
- [processingサンプルプログラムの使い方](#)
- [Androidアプリの使い方](#)

はじめに

1. [商品の概要](#)
2. [マニュアルの流れ](#)

1. 商品の概要

本製品は自律走行やBluetoothによる通信を可能にしたミニ四駆用制御ボードです。 9軸IMUセンサを搭載し、センサ情報による制御やログ取得も可能。 サンプルのアプリケーションが付属しています。



特徴

- 加減速制御のためHブリッジモータドライバを搭載
- 9軸IMU搭載
- 制御マイコンには72MHzのARM Cortex-M3を搭載
- ログ機能あり
- Bluetoothを搭載可能（Bluetoothモジュールは別売り）
- 本格的な組込機器の教材にも使用可能
- MAシャーシに対応、PROシリーズ ライキリのボディーを取付可能（ミニ四駆は別売り）

AIチップ仕様

- CPU:NXP社LPC1343
- モータドライバ:TI社製DRV8850RGYR
- センサ:9軸IMUセンサ TDK InvenSense社製MPU-9250
- 通信:Microchip社製RN42Xを搭載可能
- バッテリ:マイコン用LiPoバッテリ3.7V110mAh(付属)

- 本体モータ用単三型乾電池2本
- ※ Bluetoothモジュールおよび単三電池は別売りです。

注意

本製品を利用するにあたり必ずマニュアルを全て読んでください。

本製品の故障もしくは使用によって生じた損失等について、弊社は一切の責任を負いません。

2. マニュアルの流れ

- 初めに

この項目では、商品の概要とマニュアルの流れについて説明します。

- **AIchipの使い方**

この項目では、AIchipの仕様や、取り扱い方について説明します。

AIchipの入出力装置や、充電方法、起動方法、起動後の操作方法を記しています。

本体の使用方法に関してはこの項目を参照してください。

- **processingサンプルプログラムの使い方**

この項目では、AIチップ用に公開されているProcessingという言語で書かれたサンプルプログラムについて説明します。サンプルプログラムのダウンロード方法や、開発環境の設定方法、使用方法などはこの項目を参照してください。

- **Androidアプリの使い方**

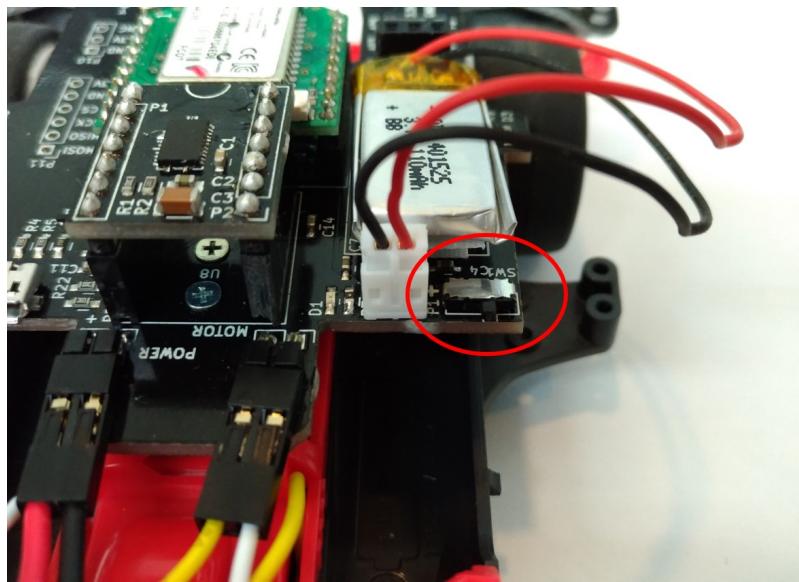
この項目では、AIチップ用に公開されているAndroid端末用サンプルアプリについて説明します。サンプルアプリのダウンロード方法や、開発環境の設定方法、アプリの使用方法などはこの項目を参照してください。

AIchipの使い方

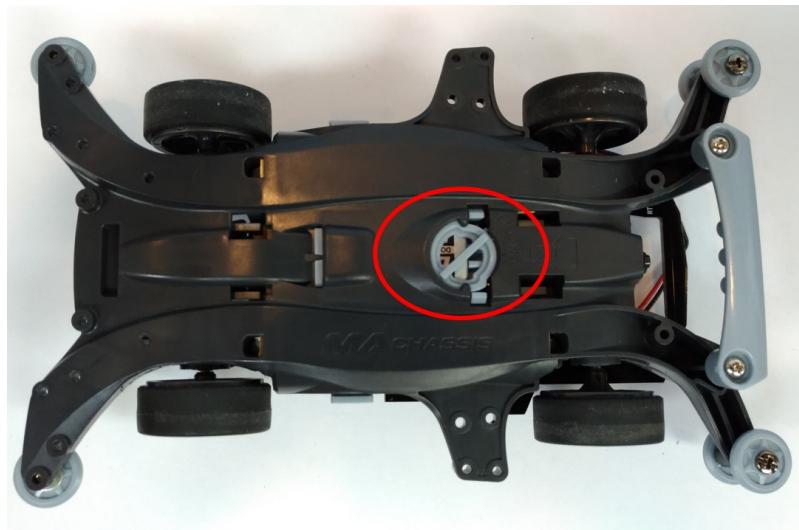
1. [AIchipの入出力装置](#)
2. [AIchipの充電方法](#)
3. [AIchipの起動方法](#)
4. [AIchipの起動モード](#)

1. AIchipの入出力装置

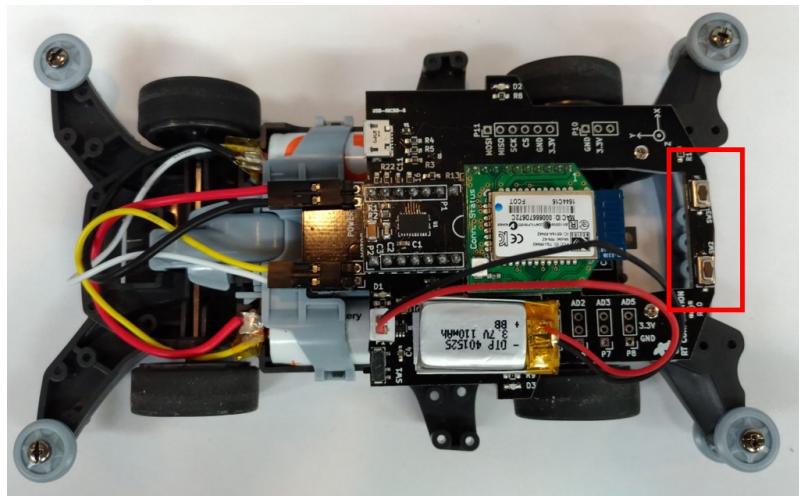
1. CPU用電源スイッチ (Fig.1-1)
2. モーター用電源スイッチ (Fig.1-2)
3. プッシュスイッチ×2 (Fig.1-3)(Fig.1-4)



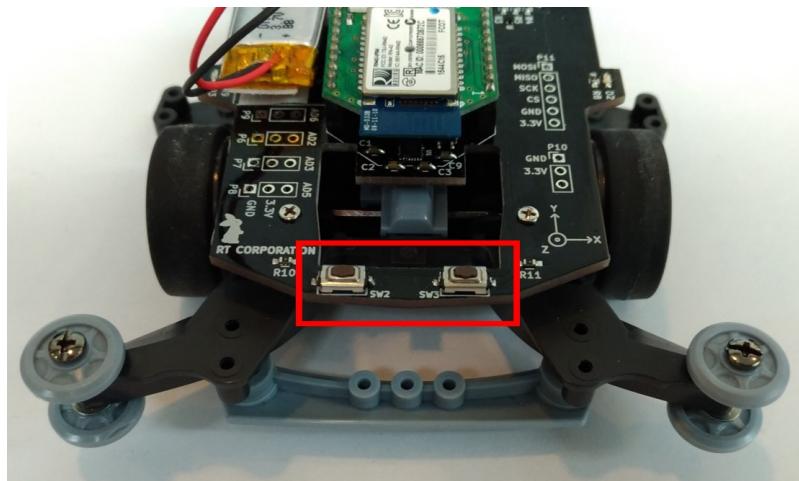
(Fig.1-1 CPU用電源スイッチ)



(Fig.1-2 モーター用電源スイッチ)



(Fig.1-3 プッシュスイッチ場所（裏から）)



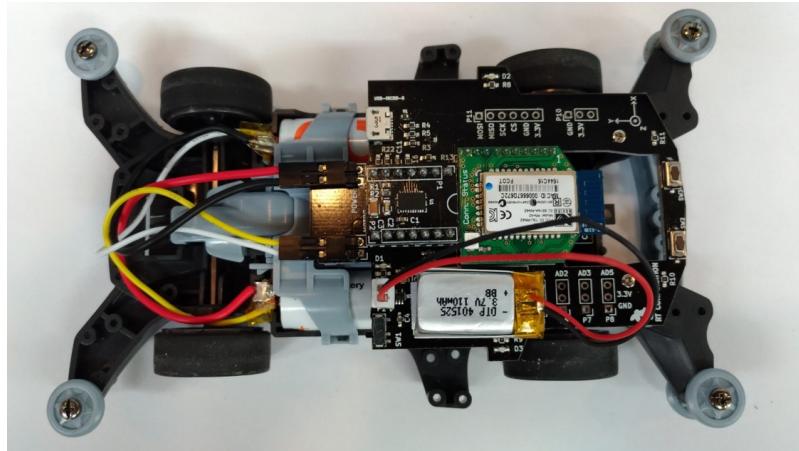
(Fig.1-4 プッシュスイッチ場所 (表から))

2. AIchipの充電方法

付属の充電器にバッテリーを繋げた後に、USBケーブルを充電器及びパソコンに接続してください。充電器のLEDが赤く光っている間は、バッテリーが充電中です。充電器のLEDが緑色に変わるとバッテリーの充電が完了します。ケーブルを持ってコネクタの抜き差しをすると、ケーブルなどの部品が破損し、大変危険です。怪我や事故の原因となりますので、バッテリーの抜き差しをする際は、必ずコネクタを持って行ってください。また、本製品はリチウムポリマー電池を使用しています。電池の異常な発熱や、バッテリーパックの膨らみを確認した際には使用を中止し、リチウムポリマー電池の廃棄方法に則り電池を廃棄してください。

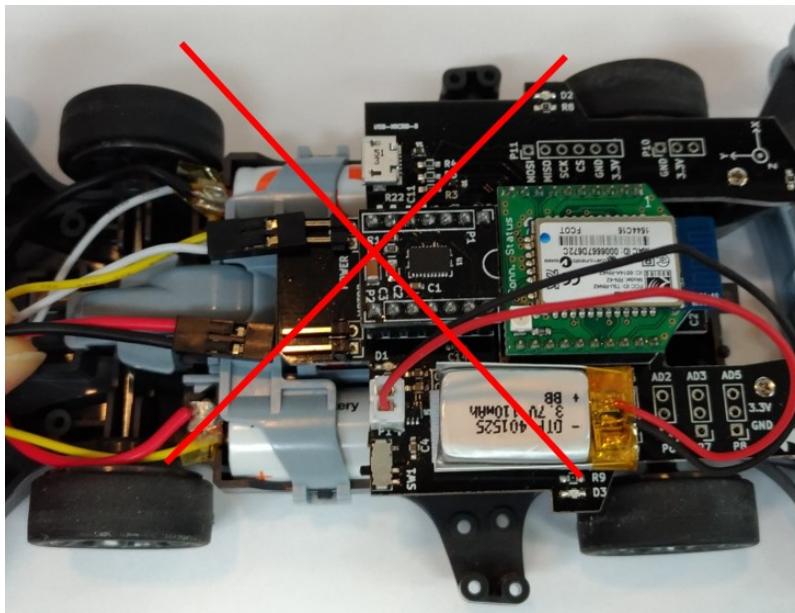
3. AIchipの起動方法

はじめに、AIchipの配線が図の通りになっているか確認をしてください。(Fig.1-5)



(Fig.1-5 通常の配線)

詳しくは「AIチップ配線マニュアル」を熟読の上、配線を行ってください。図の向きにおいていた時に、**上側がバッテリ**のケーブルが接続されていて、**下側がモータ**のケーブルが接続されているのが正しい配線です。電源の土を間違って配線をすると壊れるので絶対に逆に配線しないでください。また、下の図のように、電源とモータを間違えて配線すると壊れるので、逆に配線しないでください。(Fig.1-6)



(Fig.1-6 誤った配線)

AIchipには電源スイッチが二箇所あります。

一つが車体上部にある[CPU用電源スイッチ](#)です。

こちらはLiPoバッテリーの正面にスライドスイッチがあるので、それを「ON」または「+」側へスライドさせてください。

もう一つが、車体下部にある、[モーター用電源スイッチ](#)です。

こちらは車体の裏側についています。「ON」の角度までスイッチをひねってください。

この二つの電源をつけることで、AIchipは起動します。

4. AIchipの起動モード

AIchipには（デフォルトで）三つの起動モードが存在します。これらはCPU用電源スイッチを起動する際に切り替えることが出来ます。切り替えは車体後部についている、プッシュスイッチを用いて行います。

1. DEMOモード

このモードは何もせずにCPU用電源スイッチを入れた時に起動します。

右のプッシュスイッチを押すとモードの選択をします。選択した状態で左のプッシュスイッチを押すとモードの決定をし、各モードの動作を開始します。

各モードの動作中に左のプッシュスイッチを1秒以上長押しすると、再度モード選択が出来ます。

選択出来るモードについては以下の表に記載します。（Table.2-1）

プッシュスイッチを押す回数	緑LED発光パターン	モード
0	消灯	何なし
1	点灯	前進モード (duty最大)
2	0.5秒間隔点滅	duty選択モード
3	0.25秒間隔点滅	自律走行モード
4	0.125秒間隔点滅	Bluetoothsettingモード
5	0.062秒間隔点滅	デバッグモード

- 前進モード
右のプッシュスイッチを押すとモーターを最速で回転させます。
もう一度右のプッシュスイッチを押すと止まります。
- duty選択モード
左右のプッシュスイッチを一度押すごとにモーターの速度を変動させます。
このモードではAIchipは全後進します。
- 自律走行モード
右のプッシュスイッチを押すことでAIchipが走行を開始します。
コースに応じて速度を変動させながら走行します。
- Bluetoothsettingモード
Bluetoothの通信設定などの変更が行えるモードです。
- デバッグモード
Bluetoothからデータを出力し続けるモードです。

2. 通信モード

このモードは、車体後部左側にあるプッシュスイッチを押しながら、CPU用電源スイッチを入れた時に起動します。このモードはAIchipからデータが常に送信され、またすべてのコマンドを受け取り、動作するモードです。

各サンプルプログラムを使用する際には、このモードで起動してください。これ以外のモードを使用すると、一部機能が正常に動作しない可能性があります。

また、起動時はプッシュスイッチを緑色のLEDが二回点滅するまで必ず押し続けてください。

途中で手を離してしまうと、LEDテストモードになる可能性があるので、ご注意下さい。

3. プログラム書き込みモード プログラムを書き込む際に、このモードで起動します。

processingサンプルプログラムの使い方

1. [Processing の導入](#)
2. [サンプルプログラムのダウンロード](#)
3. [サンプルプログラムの実行](#)
4. [サンプルプログラムの解説](#)

1. Processing の導入

Processingの[ダウンロードページ](#)からProcessingをダウンロードしてください。ダウンロードの際には、自分の環境に合うOSをクリックしてください。(Fig.2-1)

The screenshot shows the official Processing Foundation website. On the left, there's a sidebar with links like Cover, Download, Donate, Exhibition, Reference, Libraries, Tools, Environment, Tutorials, Examples, Books, Handbook, Overview, People, and Shop. The main content area has a large banner for 'Processing' with a geometric background. Below the banner, it says 'Download Processing. Processing is available for Linux, Mac OS X, and Windows. Select your choice to download the software below.' A large button for '3.3.7 (13 March 2018)' is shown, with download links for Windows 64-bit, Windows 32-bit, Linux 64-bit, Linux 32-bit, and Mac OS X. Below this, there's a section for 'Stable Releases' with a list of versions: 3.3.7 (13 March 2018), 3.3.6 (4 September 2017), 2.2.1 (19 May 2014) [highlighted with a red box], 1.5.1 (15 May 2011), and an earlier removed version. A red arrow points from the Japanese text above to the '2.2.1' link. At the bottom, there's a 'Pre-Releases' section.

Download Processing. Processing is available for Linux, Mac OS X, and Windows. Select your choice to download the software below.

3.3.7 (13 March 2018)

[Windows 64-bit](#) [Linux 64-bit](#) [Mac OS X](#)
[Windows 32-bit](#) [Linux 32-bit](#)
[Linux ARMv6hf](#)

» [Github](#) [Report Bugs](#) [Wiki](#) [Supported Platforms](#)

Read about the [changes in 3.0](#). The list of revisions covers the differences between releases in detail.

Stable Releases

- 3.3.7 (13 March 2018) Win 32 / Win 64 / Linux 32 / Linux 64 / Linux ARMv6hf / Mac OS X
- 3.3.6 (4 September 2017) Win 32 / Win 64 / Linux 32 / Linux 64 / Linux ARMv6hf / Mac OS X
- 2.2.1 (19 May 2014) Win 32 / Win 64 / Linux 32 / Linux 64 / Mac OS X
- 1.5.1 (15 May 2011) Win (standard) / Win (no Java) / Linux x86 / Mac OS X

Earlier releases have been removed because we can only support the current versions of the software. To update old code, read the [changes](#) page. Changes for each release can be found in [revisions.txt](#). If you have problems with the current release, please [file a bug](#) so that we can fix it. Older releases can also be [built from the source](#). [Read More](#) about the releases and their numbering. To use [Android Mode](#), Processing 3 or later is required.

Pre-Releases

(Fig.2-1 ダウンロード画面1)

過去のバージョンをダウンロードしたい場合は、画面中部にある「Stable Release」の項目からダウンロードしてください。こちらでダウンロードする場合も、お使いのOSが表記されているリンクを選択してダウンロードしてください。

現在サンプルアプリケーションの動作が確認できているバージョンは **3.0 beta 4** 及び **2.2.1**です。Processingをダウンロードする際にはバージョンにご注意下さい。ダウンロードができたら、ソフトのインストールを行ってください。

2. サンプルプログラムのダウンロード

ProcessingのサンプルプログラムはGitHubの[AI-CHIPV3のページ](#)から行ってください。

ダウンロードする際には、画面右側の「Clone or download」をクリックしてください。(Fig.2-4)

No description, website, or topics provided.
Add topics

9 commits 2 branches 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

kenjinukai Initial commit Latest commit 1f59628 18 hours ago

Android/BloothoothChat Initial commit 19 hours ago

MCUXpresso プログラムアップデート 20 hours ago

Processing Initial commit 18 hours ago

circuit 回路について追記 20 hours ago

firmware V3用プログラム&ファームウェアUP 22 hours ago

README.md 回路について追記 20 hours ago

README.md

AlchipV3

(Fig.2-4 githubのページ) ZIPファイルがダウンロード出来たら、ダウンロードファイルを任意の場所に解凍してください。

3. サンプルプログラムの実行

まず、[1. Processing の導入](#)でインストールしたProcessingを起動してください。

processingを起動したら、[2. サンプルプログラムのダウンロード](#)で解凍したフォルダ内の「Processing」→「sample_App」内にある「sample_App.pde」を開いてください。

開いたら、パソコンとAIchipのBluetoothのペアリングと通信ポートの設定を行ってください。 (Fig.2-5)



```

1 //シリアル通信ライブラリを取り込む
2 import processing.serial.*;
3 //ポートのインスタンス
4 Serial port;
5 // シリアルポートから取得したデータ(Byte)
6 int inByte;
7 float m_duty = 0;
8
9 void setup(){
10   //画面の設定
11   size(100,100);
12   //シリアルポート設定 (Bluetoothのポート)
13   port=new Serial(this,"/dev/tty.RNBT-4EA9-RNI-SPP",115200); //Mac
14   //port=new Serial(this,"COM5",115200); //Windows
15   port.write(command0(0));

```

(Fig.2-5 サンプルプログラムのコード)

- Windows

コントロールパネルを開いて、ハードウェアとサウンドにある項目の「デバイスの追加」を選択します。(Fig.2-6)

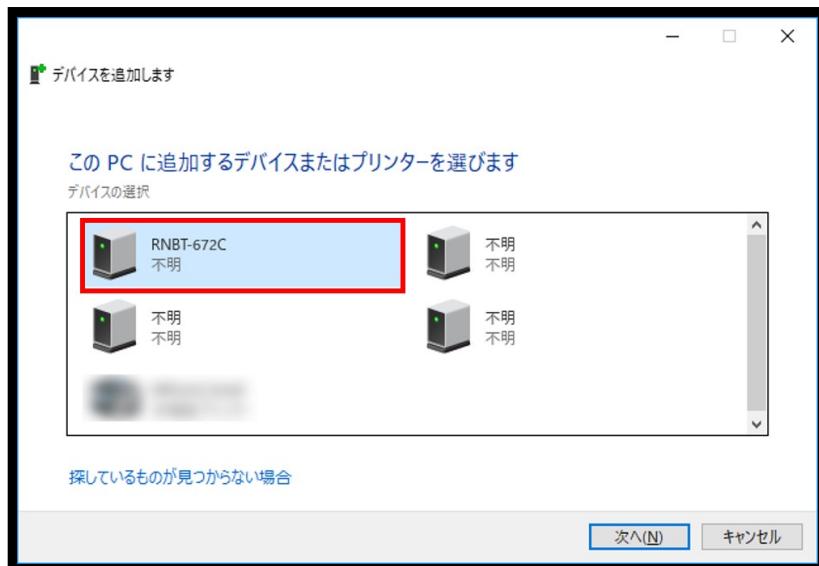


(Fig.2-6 コントロールパネルの画面)

デバイスの追加の画面が表示されたら、AIchipを通信モードで起動します。

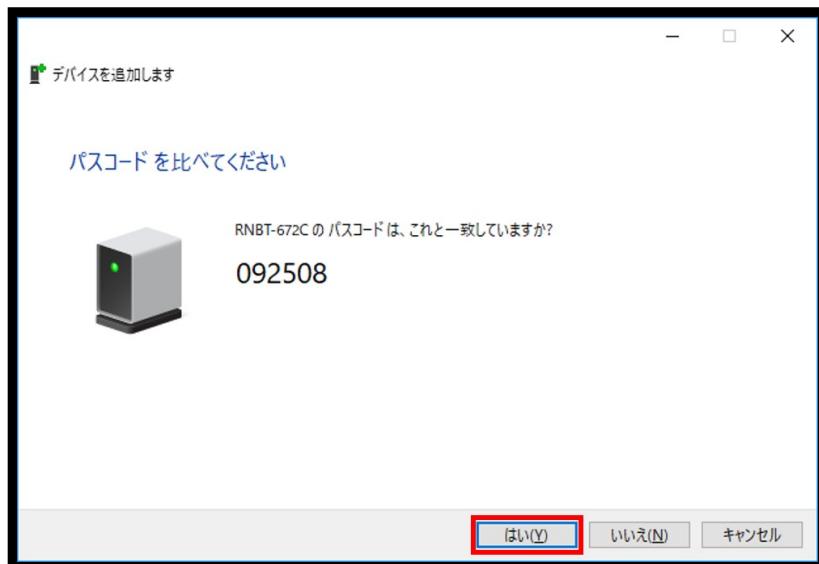
起動すると、通常Bluetoothモジュールが赤く点滅します。

パソコン側で認識すると、「RNBT-xxxx」が表示されるので、選択して「次へ」を押してください。この時選択した名前は、後ほど使うので覚えておいてください。(Fig.2-7)



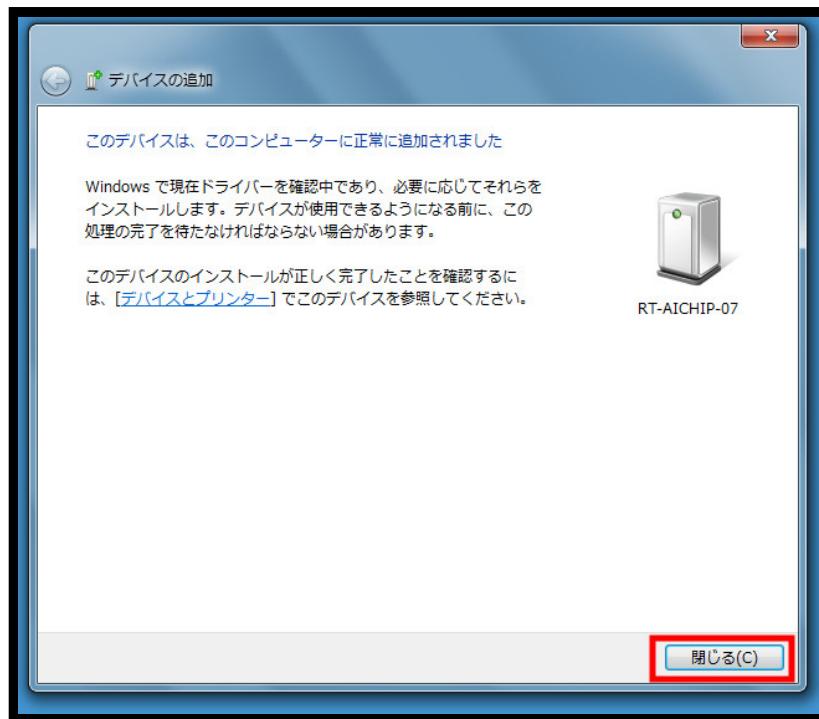
(Fig.2-7 デバイスの選択画面)

次に、ペアリングコードの比較のウィンドウが表示されるので、「次へ」を押してください。(Fig.2-8)



(Fig.2-8 ペアリングコード比較画面)

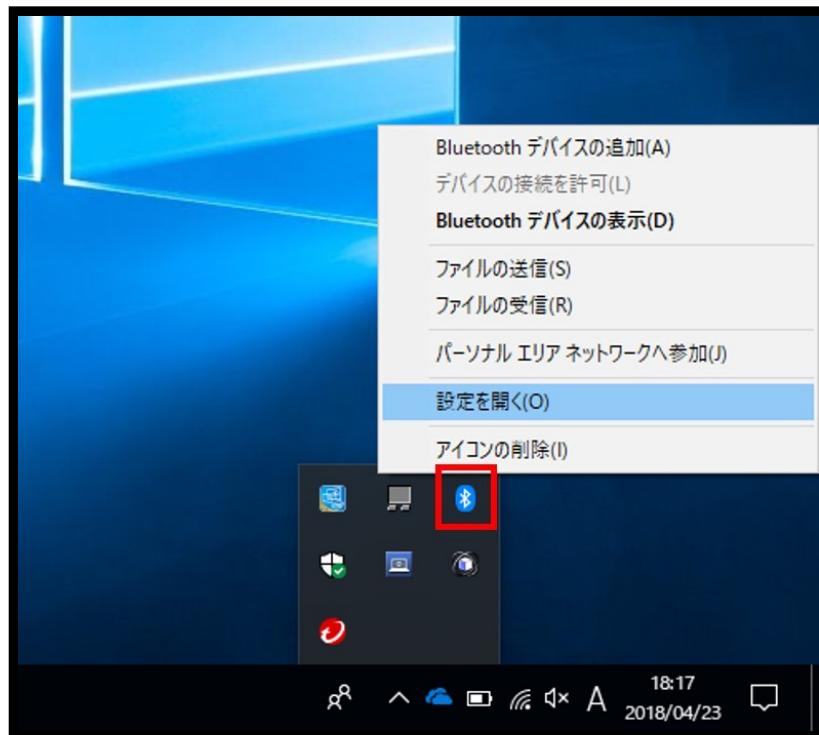
最後に次のような画面が表示されると接続が完了します。(Fig.2-9)



(Fig.2-9 接続完了画面)

接続が完了したら、接続されたポートの確認を行います。ポートの確認は、Bluetoothの設定画面から行います。

まず、画面右下にあるタスクトレイの中の、Bluetoothアイコンを右クリックしてください。右クリックをするとメニューリストが開かれるので「設定を開く」を選択してください。(Fig.2-10)



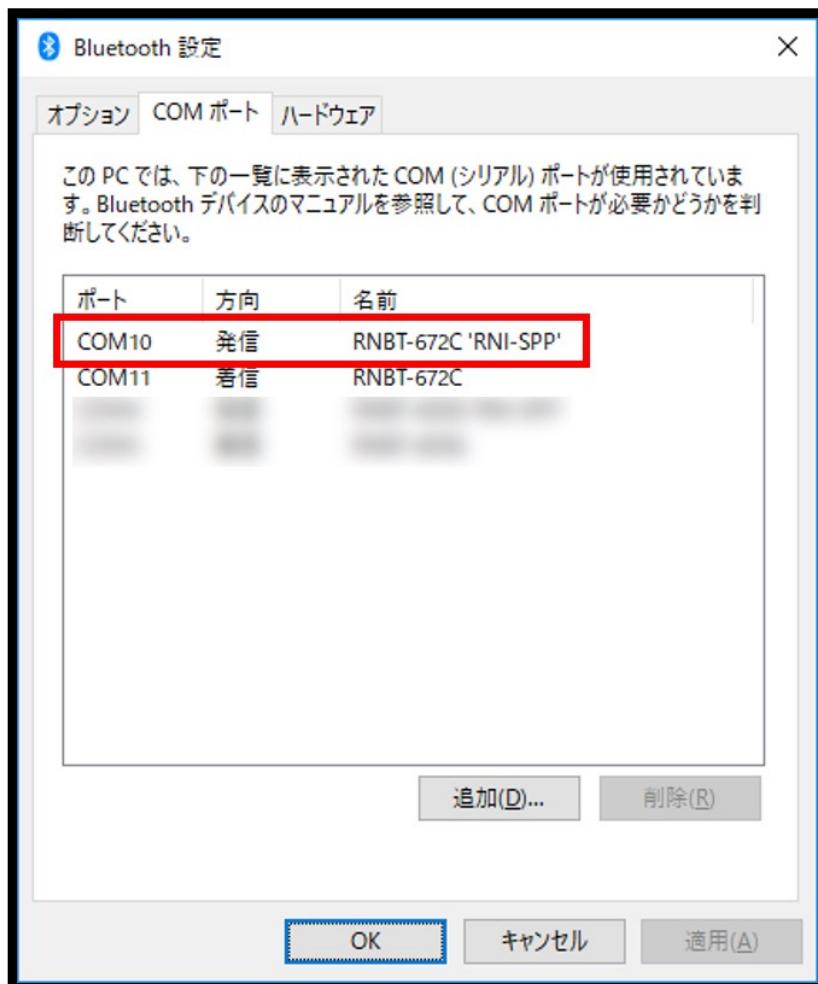
(Fig.2-10 Bluetoothアイコンの選択)

「設定を開く」をクリックすると、次のような「Bluetooth」の設定画面が開かれます。開かれたら、タブの「COMポート」をクリックしてください。(Fig.2-11)



(Fig.2-11 Bluetoothの設定画面)

「COMポート」をクリックすると、パソコンに接続されたBluetooth機器の一覧が表示されます。この中の、先ほど接続した名前と「'RNI-SPP'」が書かれた欄を確認してください。
方向が「発信」となっているCOMポートを確認してください。(Fig.2-12)

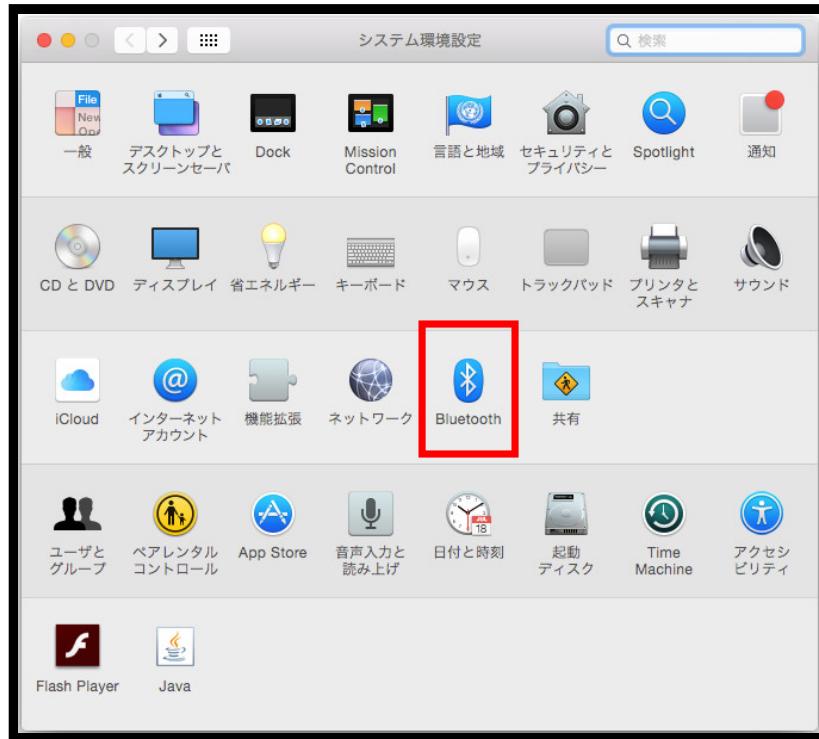


(Fig.2-12 COMポート確認画面)

確認したポートはプログラムの14行目に書き込みます。デフォルトで「"COM4"」となっている部分を自分で確認したCOMポートに書き換えてください。

- Mac

システム環境設定のBluetoothを選択します。次に、AIchipの電源を、通信モードでつけます。
(Fig.2-13)



(Fig.2-13 システム環境設定画面)

その後、システム環境設定ウィンドウ内に表れるRNBT-××××を選択し、ペアリングを行います。(Fig.2-14)



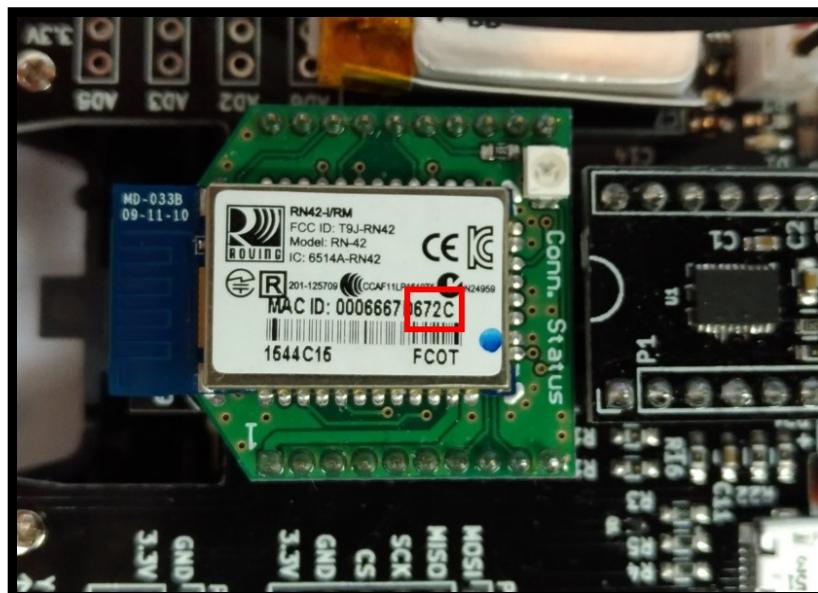
(Fig.2-14 Buletooth接続画面)

接続が完了すると、「接続済み」の表記になります。(Fig.2-15)



(Fig.2-15 Bluetooth接続確認)

この「xxxx」の部分は、BluetoothモジュールのMAC IDの下四桁になります。
各自確認して、プログラムを変更してください。(Fig.2-16)

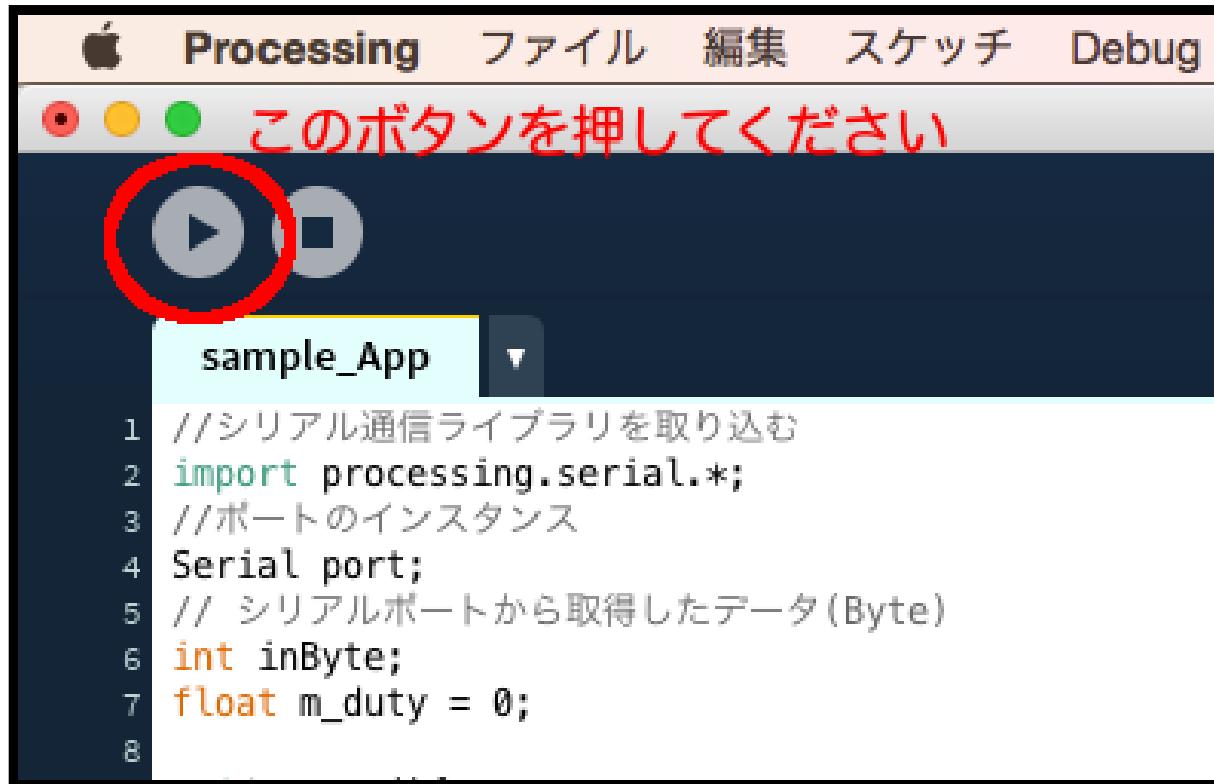


(Fig.2-16 BluetoothモジュールのID確認)

ウィンドウのデバイス欄に表示されたデバイス名が違う場合は、13行目の設定を書き換えてください
「"/dev/tty.RNBT-xxxx-RNI-SPP"」

デフォルトでは13行目がコメントアウト（プログラムに反映されない状態に）されています。13行目の先頭の「//」を削除し、14行目の先頭に「//」を書き足してください。

ペアリングと通信ポートの設定出来たら、processingのプログラムを起動させてください。 (Fig.2-17)



(Fig.2-17 processingの実行ボタン)

起動するとウィンドウが一つ立ち上がります。ウィンドウ上でマウスクリックをすると、AIchipの縁のLEDが点灯します。

また、エディタ下部にはログ出力が表示されます。ここにはAIchipからのデータが出力されます。データの詳細は以下をご覧ください。(Table.2-1)(Table.2-2)

(Table.2-1)

受信データのプロトコル1			
Byte	内容	Byte	内容
0	0xff	11	ACC Y 上位8bit (符号付)
1	0xff	12	ACC Z 下位8bit (符号付)
2	0x52	13	ACC Z 上位8bit (符号付)
3	0x54	14	TEMP 下位8bit (符号付)
4	0x34	15	TEMP 上位8bit (符号付)
5	0x57	16	GYRO X 下位8bit (符号付)
6	0x00	17	GYRO X 上位8bit (符号付)
7	タイムスタンプ	18	GYRO Y 下位8bit (符号付)
8	ACC X 下位8bit (符号付)	19	GYRO Y 上位8bit (符号付)
9	ACC X 上位8bit (符号付)	20	GYRO Z 下位8bit (符号付)
10	ACC Y 下位8bit (符号付)	21	GYRO Z 上位8bit (符号付)

(Table.2-2)

受信データのプロトコル2

Byte	内容	Byte	内容
22	MAG X 下位8bit (符号付)	33	isCurve (符号なし)
23	MAG X 上位8bit (符号付)	34	isSlope (符号なし)
24	MAG Y 下位8bit (符号付)	35	経過時間 0byte (符号なし)
25	MAG Y 上位8bit (符号付)	36	経過時間 1byte (符号なし)
26	MAG Z 下位8bit (符号付)	37	経過時間 2byte (符号なし)
27	MAG Z 上位8bit (符号付)	38	経過時間 3byte (符号なし)
28	角度 下位8bit (符号付)	39	Lipo電圧 下位8bit (符号なし)
29	角度 上位8bit (符号付)	40	Lipo電圧 上位8bit (符号なし)
30	duty 下位8bit (符号付)	41	モーター電圧 下位8bit (符号なし)
31	duty 上位8bit (符号付)	42	モーター電圧 上位8bit (符号なし)
32	isStop (符号なし)		

24

4. サンプルプログラムの解説

- サンプルプログラムの機能
- サンプルの流れ
- コマンドの生成方法

サンプルプログラムの機能

プログラムを実行すると、ウィンドウが一つ開きます。そのウィンドウ上をクリックすると、AIchipの緑のLEDが点灯します。

エディタ下部の、ターミナルにAIchipからの出力データが表示されます。このデータは (Table.2-1) と (Table.2-2) を参考に解析してください。

また、キーボードのArrowkeyの右と左を押すとAIchipのモーターが動きます。

サンプルプログラムは以上の3つの機能を有しています。

サンプルの流れ

はじめに次の行が実行されます

```
//シリアル通信ライブラリを取り込む
import processing.serial.*;
//ポートのインスタンス
Serial port;
//シリアルポートから取得したデータ(Byte)
int inByte;
float m_duty = 0;
```

ここではプログラムの実行に必要なライブラリの取り込みや、Bluetoothの通信に必要なインスタンスの生成及びグローバル変数の定義を行っています。

この行はプログラム開始時に一回だけ実行されます。

また、この処理が終了すると、各関数が初めて実行されるようになります。

```
void setup(){
//画面の設定
size(100,100);
//シリアルポート設定 (Bluetoothのポート)
//port=new Serial(this,"/dev/tty.RNBT-4EA9-RNI-SPP",115200); //Mac
port=new Serial(this,"COM4",115200); //Windows
port.write(command0(0));
}
```

setup関数は、プログラムが開始すると初めに一度だけ実行される関数です。

この関数では、実行時に出てくるウィンドウのサイズの設定と、通信ポートの設定を行っています。

必要に応じて設定内容の変更を行って下さい。詳しい設定の方法は [3. サンプルプログラムの実行](#)に記載されています。

うまく動作しない場合はそちらを確認してください。

```
void draw(){ background(51); }
```

draw関数はループして実行されます。

background関数は表示されるウィンドウの背景の色を変更しているだけです。

```
void serialEvent(Serial p){
// 設定したシリアルポートからデータを読み取り
inByte = port.read();
println(hex(inByte));
}
```

serialEvent関数はAIchipからデータが、送られるたびに実行されます。

データはprintln関数により、ターミナルに表示されます。

```
void keyPressed() {
if (key == CODED) { // コード化されているキーが押された
if (keyCode == LEFT) {
if(m_duty > -1){
m_duty -= 0.1;
//println(m_duty);
}
port.write(command0(m_duty));
}else if(keyCode == RIGHT){
if(m_duty < 1){
m_duty += 0.1;
//println(m_duty);
}
port.write(command0(m_duty));
}
}
}
```

keyPressed関数はキーボードが押されるたびに実行されます。今回はArrowkeyの左と右が押された時のみ有効になっています。

コマンドはcommand0関数を用いて生成されます。
生成されたコマンドはport.write関数を用いて、AIchipに送信されます。
コマンドの生成方法については後述します。

```
/**

- id 0: dutyの変更コマンド
- 
- @param duty dutyを-1.0から1.0で指定



- 負の値はモーターを逆の方向に回す



- @return 10byteのコマンド配列



```
*/
byte[] command0(float duty){
 byte[] command = new byte[10];
 int int_duty;
 int duty_L;
 int duty_H;

 int_duty = (int)(duty * 32767.0);
 if(int_duty<0)
 {
 int_duty += 65535;
 }
 duty_L = int_duty & 0x000000ff ;
 duty_H = (int_duty & 0x0000ff00)>>8;

 //ヘッダー
 command[0] = 99;
 command[1] = 109;
 command[2] = 100;
 //id
 command[3] = 0;
 //値
 command[4] = byte(duty_L);
 command[5] = byte(duty_H);
 //ダミー
 command[6] = 0;
 command[7] = 0;
 command[8] = 0;
 command[9] = 0;

 return command;
}
```


```

このコマンドはkeyPressed内で実行されます。

引数のdutyによってAIchipのモーターのパワーを変更するようなコマンドを生成します。

```
void mousePressed(){
byte[] command = new byte[10];
//ヘッダー
command[0] = 99;
command[1] = 109;
command[2] = 100;
//id
command[3] = 1;
//値
command[4] = byte(1);
//ダミー
command[5] = 0;
command[6] = 0;
command[7] = 0;
command[8] = 0;
command[9] = 0;
port.write(command);
}
```

mousePressed関数はマウスがクリックされると実行されます。

今回は緑のLEDが点灯するようにコマンドを生成したのちに、AIchipに送信しています。

```
void mouseReleased(){
byte[] command = new byte[10];
//ヘッダー
command[0] = 99;
command[1] = 109;
command[2] = 100;
//id
command[3] = 1;
//値
command[4] = byte(0);
//ダミー
command[5] = 0;
command[6] = 0;
command[7] = 0;
command[8] = 0;
command[9] = 0;
port.write(command);
}
```

mouseRelease関数はマウスのクリックが解除されると実行されます。

今回は緑のLEDが消灯するようにコマンドを生成したのちに、AIchipに送信しています。

コマンドの生成方法

AIchipへの送信コマンドは要素数10のbyte配列を使用します。（固定長）(ref.2-1)

ヘッダーの始め3byteは必ず固定で、「99」「109」「100」を代入してください。4byte目はコマンドidです。この数字を変更することで、AIchipへの命令の種類を変えることができます。

5byte目以降はデータフィールドです。データフィールドは全て利用するわけではありません。必要のない要素には「0」を代入してください。詳細は、同ディレクトリ内に存在する「AIchipハックソン」を参照ください。

送信コマンドのプロトコル

- 長さ10byteで以下のフォーマット

0byte	1byte	2byte	3byte	4byte	5byte	6byte	7byte	8byte	9byte
header			id	datafield					
99	109	100	XX	XX	XX	XX	XX	XX	XX
固定			コマンド毎に異なる						

- 意味のある10byteコマンドをAICHP側が受信すると対応した動作を実行
- 送信コマンドはid 0,1, ..., 5の5種類
- BluetoothChatのbyte列送信関数をもじいて10byteのコマンドを送る

13

(ref.2-1 コマンド送信プロトコルの説明)

- id = 0** モーターコントロール用コマンド

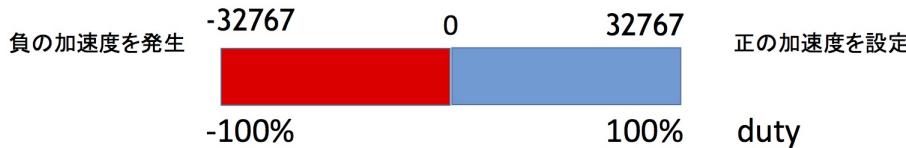
モーターを制御するには、送信コマンドの4byte目の「id」を「0」にする必要があります。(ref.2-2)
データフィールドは5byte目と6byte目にdutyの値を入れます。その他のデータフィールドには「0」を代入します。

dutyの値は16bitの符号付整数です。上位bitが5byte目、下位bitが6byte目です。配列の順番に気をつけてください。

モーターduty設定コマンド

0byte	1byte	2byte	3byte	4byte	5byte	6byte	7byte	8byte	9byte
header			id	datafield					
99	109	100	0	XX	XX	0	0	0	0

16bit符号付整数 ←ここを与えたいdutyに応じて変更
アクセルの吹かし具合



速度ではなく加速度を操作していることに注意

14

(ref.2-2 モーター制御コマンド)

- ***モーターのduty設定例***
- ***duty=50%***のコマンド例

```
> /*

int_duty = (int)(0.5 /* 32767.0);
int_duty = 16384 正確には16383.5ですが、切り上げました。
16384を16進数に変換すると0x4000になります。
よって duty_L = 00, duty_H = 40 となります。
*/
//ヘッダー
command[0] = 99;
command[1] = 109;
command[2] = 100;
//id
command[3] = 0;
//値
command[4] = byte(00);
command[5] = byte(40);
//ダミー
command[6] = 0;
command[7] = 0;
command[8] = 0;
command[9] = 0;
```

- ***duty=20%***のコマンド例

```
> /*

int_duty = (int)(0.2 /* 32767.0);
int_duty = 6553 正確には6553.4ですが、切り下げました。
16384を16進数に変換すると0x1999になります。
よって duty_L = 99, duty_H = 19 となります。
*/
```

```
//ヘッダー
command[0] = 99;
command[1] = 109;
command[2] = 100;
//id
command[3] = 0;
//値
command[4] = byte(99);
command[5] = byte(19);
//ダミー
command[6] = 0;
command[7] = 0;
command[8] = 0;
command[9] = 0;

- ***duty=-50%***のコマンド例
> /*

int_duty = (int)(-0.5 /* 32767.0);
int_duty = -16384 正確には-16383.5ですが、切り下げました。
int_dutyが負の値なので65535を加えます。加えた値は次のようになります。
int_duty = (-16384 + 65535) = 49151
49151を16進数に変換すると0x0000BFFFになります。
よって duty_L = FF, duty_H = BF となります。
*/
//ヘッダー
command[0] = 99;
command[1] = 109;
command[2] = 100;
//id
command[3] = 0;
//値
command[4] = byte(FF);
command[5] = byte(BF);
//ダミー
command[6] = 0;
command[7] = 0;
command[8] = 0;
command[9] = 0;
```

- **id=1** 緑のLED制御コマンド

LEDを制御するには、送信コマンドの4byte目の「id」を「1」にする必要があります。(ref.2-3)
 データフィールドは5byte目に「0」を入れると、緑のLEDが消灯します。「1」を入れると緑のLED
 が点灯します。
 その他のデータフィールドは「0」を代入します。

LEDの点灯制御コマンド

- 右(緑)LEDの制御コマンド

0byte	1byte	2byte	3byte	4byte	5byte	6byte	7byte	8byte	9byte
header			id	datafield					
99	109	100	1	XX	0	0	0	0	0

- 左(赤)LEDの制御コマンド 1:LED点灯 0:LED消灯

0byte	1byte	2byte	3byte	4byte	5byte	6byte	7byte	8byte	9byte
header			id	datafield					
99	109	100	2	XX	0	0	0	0	0

1:LED点灯 0:LED消灯

16

(ref.2-3) LED制御コマンド

Androidアプリの使い方

1. Androidアプリのインストールから導入まで
2. Androidサンプルアプリの使い方
3. Androidアプリに付属しているライブラリの使いかた

1. Androidアプリのインストールから導入まで

インストール方法としては2通りの方法があります。

1つ目は、パッケージファイルから直接インストールする方法です。ダウンロードした「AIchipV3」フォルダから「Android」→「BluetoothChat」→「Application」→「build」→「outputs」→「apk」の中の「Application-debug.apk」をAndroid端末に移動します。Android端末の設定から「セキュリティ」→「提供元不明のアプリ」のインストールを許可します。Android端末から「Application-debug.apk」を直接起動することでインストールできます。

2つ目は、Androidアプリの開発環境である「Android Studio」からAndroid端末にインストールする方法です。まずAndroid開発環境である「Android Studio」を、[AndroidStudioのダウンロードページ](#)からダウンロードしてください。「DOWNLOAD OPTION」に各OSごとのインストールパッケージが置いてあるので、そこからダウンロードしてください。(Fig.3-1)

Platform	Android Studio package	Size	SHA-256 checksum
Windows (64-bit)	android-studio-ide-173.4819257-windows.exe Recommended	758 MB	2d11cd16fffc7f4acaf2bd95b000cafb49a854a07ba1a353ad765fc102ae9b
	android-studio-ide-173.4819257-windows.zip No .exe installer	655 MB	6941761a932499bd9c0f5d7548ff16fcfc65c9e71ea70e1bc75d066b51b77c7dd
Windows (32-bit)	android-studio-ide-173.4819257-windows32.zip No .exe installer	854 MB	85cbfb033c941b3abcb70caeac34ea034214a079376fb1b0e10bc7b4ed71c05cf0
Mac	android-studio-ide-173.4819257-mac.dmg	849 MB	d4a8502c5aabfc5477ff30dfffe296bf705bd7e62650a76796b646a8f2805e5c
Linux	android-studio-ide-173.4819257-linux.zip	853 MB	d86748e44d658fd39581b40f7b706fb397fc1eca5dd5f8066a56c0be8856dea4

(Fig.3-1 Android Studioのダウンロードページ)

次にダウンロードしたファイルから、「Android Studio」をインストールしてください。インストールが終了したら、次はAIchipV3のgithubのページから、プロジェクトをダウンロードします。ダウンロードする際には、画面右側の「Clone or download」をクリックしてください。(Fig.3-2)

rt-net / AlchipV3

No description, website, or topics provided.

Add topics

9 commits 2 branches 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

kenjiinukai Initial commit Latest commit 1f59628 18 hours ago

Android/BloothoothChat Initial commit 19 hours ago

MCUXpresso プログラムアップデート 20 hours ago

Processing Initial commit 18 hours ago

circuit 回路について追記 20 hours ago

firmware V3用プログラム&ファームウェアUP 22 hours ago

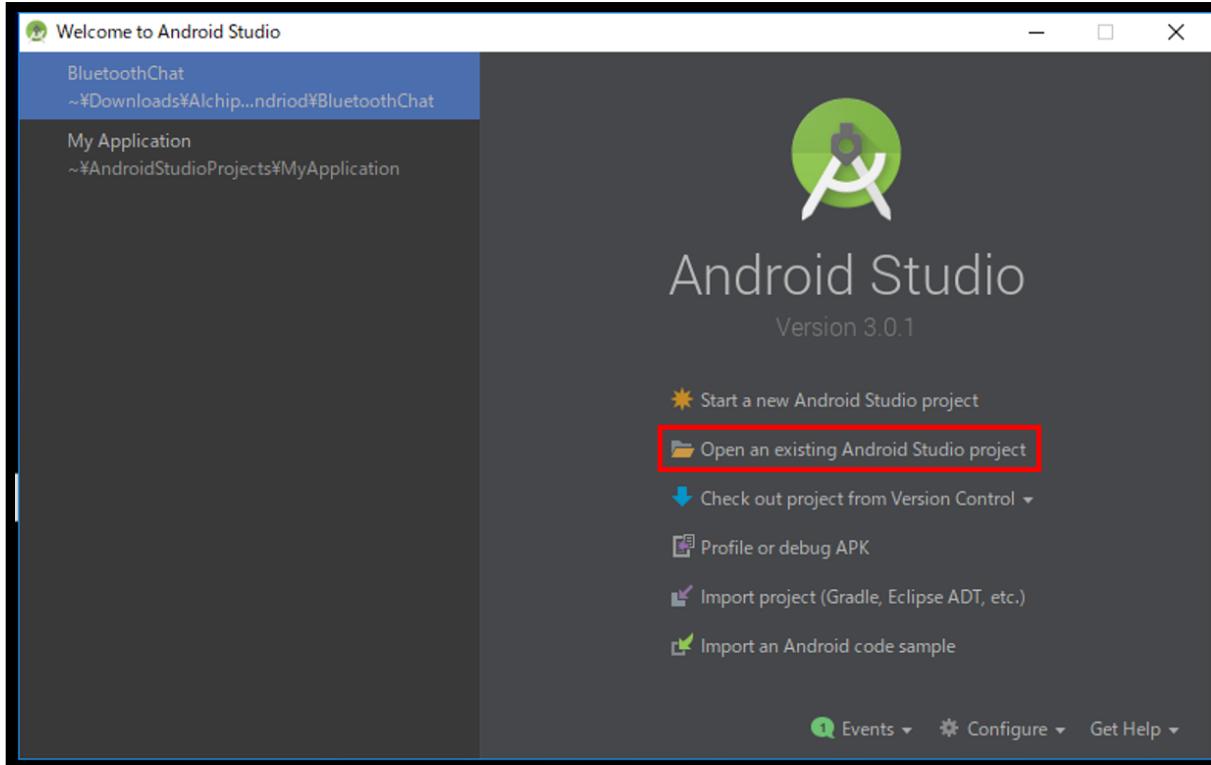
README.md 回路について追記 20 hours ago

README.md

AlchipV3

(Fig.3-2 githubのページ)

ダウンロードが完了したら、ダウンロードしたZIPファイルを任意の場所に解凍してください。
ここまで完了したら、「Android Studio」を起動してください。起動すると、次のような画面が開かれると
思うので、「Open an existing Android Studio project」を選択してください。(Fig.3-3)

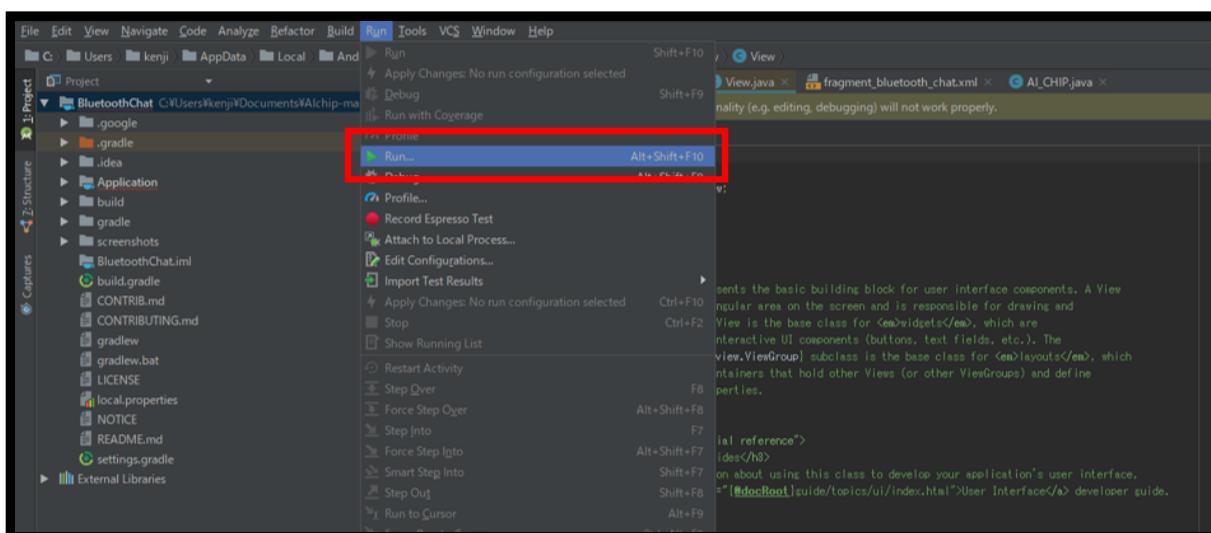


(Fig.3-3 Android Studioの起動画面)

選択したら、先ほど解凍した「AIchipV3」ディレクトリ内にある「BluetoothChat」を選択して、開いてください。このプロジェクトが、AIchipのAndroid端末用サンプルアプリのプロジェクトです。

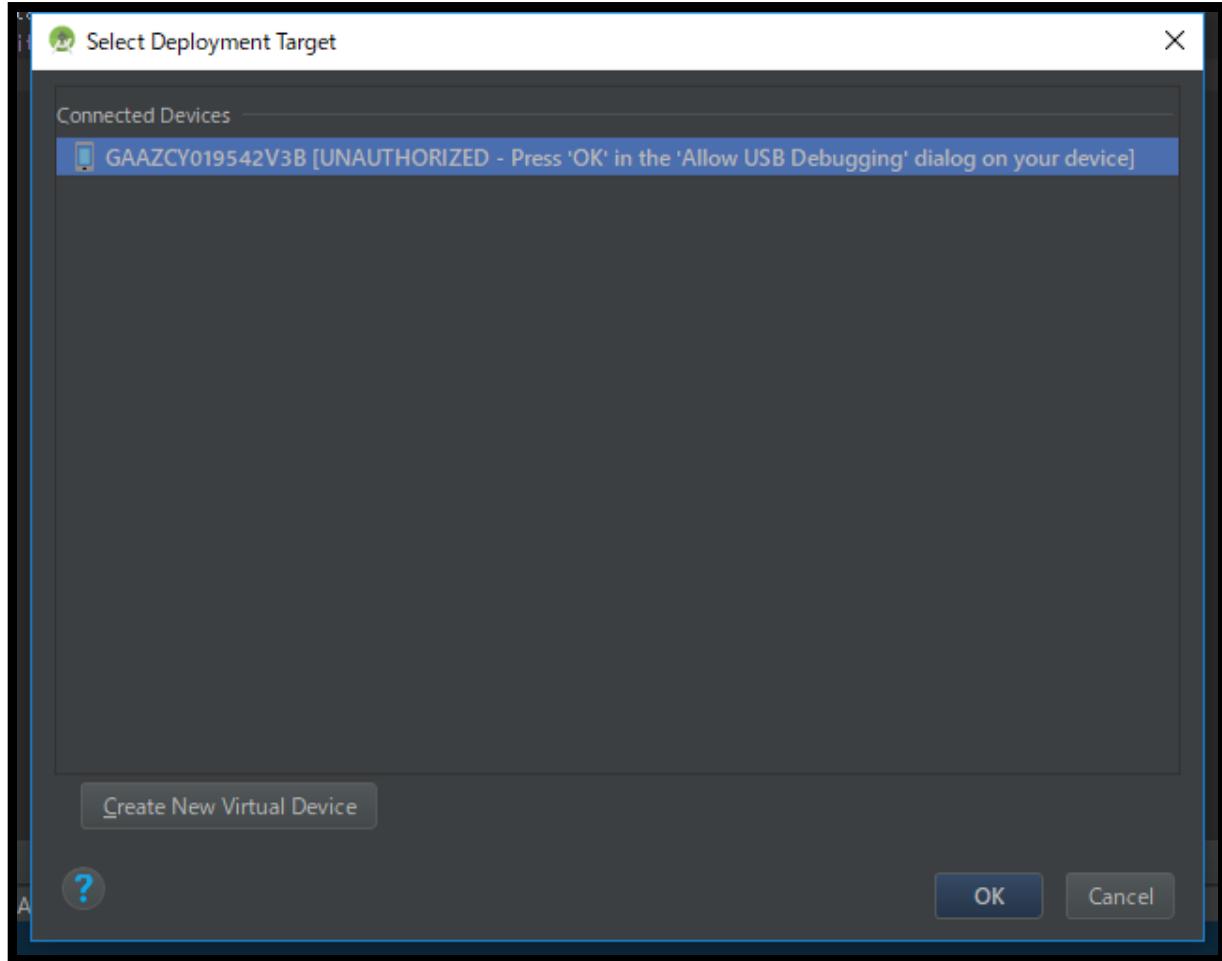
次に、サンプルアプリをAndroid端末に転送する手順です。お手持ちのAndroid端末を「USBデバッグモード」をONにしてPCと接続してください。

接続ができたら、Android Studio上部にある「Run」の中から緑の三角アイコン「Run」をクリックしてください。(Fig.3-4)



(Fig.3-4 プログラムの転送1)

「Connected Devices」にお手持ちのデバイスが表示されていれば書き込み可能です。(Fig.3-5)

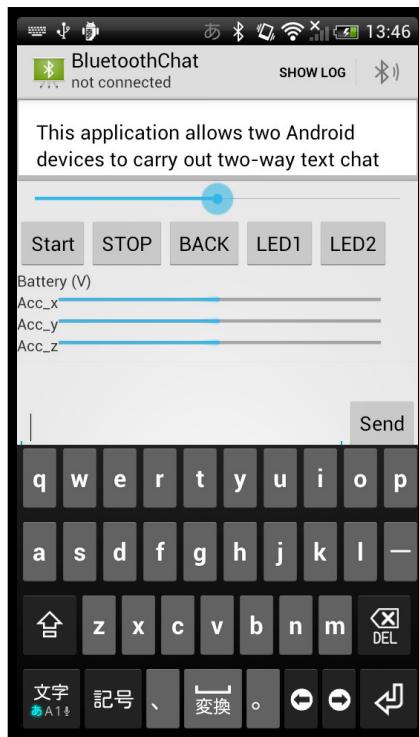


(Fig.3-5 プログラム転送2)

「OK」を選択して、書き込みをしてください。お手持ちのデバイスに「BluetoothChat」のアプリがインストールされていれば成功です。

2. Andoridサンプルアプリの使い方

画面上にはスイッチが五つとプログレスバーが三つと、シークバーが配置されており、それぞれがAIchipと情報の送受信が可能です。(Fig.3-6)



(Fig.3-6 Androidアプリ画面)

AIchipと通信をするためには、画面右上のBluetoothアイコンをおして、ペアリングをしてください。接続ができると、画面上の「non connection」が接続したデバイス名になります。画面中央に配置された、左三つのスイッチを押すとAIchipのモーターが動きます。右側二つのスイッチを押すとLEDが点灯します。シークバーをスライドさせると、AIchipの速度が変動します。

画面上のプログレスバーは、それぞれx,y,z,軸方向のAIchipの加速度を示します。

Battery(V)の部分にはリアルタイムでAIchipのモーター電圧が表示されます。

文字を何も打たずに「send」ボタンを押すと、音声入力モードになり、特定のキーワードに反応してAIchipが動きます。(Table.3-1)

特定の文字を打ち込んで「send」ボタンを押すと、AIchipが動きます。(Table.3-2)

(Table.3-1 音声入力で反応するキーワード)

音声入力	AIchipの動作
動け、行け、Go!	前進100%
ゆっくり	前進50%
ストップ、止まれ	停止
戻れ、バック	後進100%

(Table.3-2 テキスト入力で反応するキーワード)

テキスト入力	AIchipの動作
0	停止
set	ジャイロセンサー初期化
get	ジャイロセンサー値取得

テキスト入力	AIchipの動作
gon	緑色LED点灯
goff	緑色LED消灯
gf	緑色LED点滅
ron	赤色LED点灯
roff	赤色LED消灯
rf	赤色LED点滅

3. Androidアプリに付属しているライブラリの使いかた

BluetoothChatService.javaを継承した、AI-CHIP.javaがAIchipのライブラリーです。 AIchipからのデータのget関数と、AIchipへのset関数が用意されています。 Androidアプリ内で関数を呼び出して利用してください。 詳しくはAI-CHIP.javaを参照してください。