AI Chip V3 マニュアル



株式会社アールティ 2018 年 7 月

目次

目次	2
1. はじめに	3
1.1 製品の概要	3
1.2 マニュアルの流れ	4
2. Al Chip の使い方	4
2.1 Al Chip の入出力装置	4
2.2 Al Chip の充電方法	5
2.3 Al Chip の起動方法	5
2.4 Al Chip の起動モード	6
3. Processing サンプルプログラムの使い方	7
3.1 Processing の導入	7
3.2 サンプルプログラムのダウウンロード	8
3.3 サンプルプログラムの実行	8
3.4 サンプルプログラムの解説	
4. Android アプリの使い方	17
4.1 Android アプリのインストールから導入まで	
4.2 Android サンプルアプリの使い方	
4.3 Android アプリに付属しているライブラリの使い方	
改版履歴	
製造元	20
製品に関するお問い合わせ	20

1. はじめに

1.1 製品の概要

本製品は自律走行や Bluetooth(オプション製品)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。



※ミニ四駆本体と Bluetooth モジュールは別売りです。

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。モジュール化がされており、故障時も交換可能。
- * 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

AI Chip V3(以下 AI Chip)の仕様

- * 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 推奨 Bluetooth モジュール: Microchip 社製 RN42XVP-I/RM, 秋月電子通商 AE-RN-42-XB
- * 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。

1.2 マニュアルの流れ

はじめに

この項目では、製品の概要とマニュアルの流れについて説明します。

Al Chip の使い方

この項目では、Al Chip の仕様や、取り扱い方について説明します。 Al Chip 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 サンプルの使い方

この項目では、AI Chip 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。 ※processing により AI Chip と通信を行う際は、Bluetooth モジュールをご購入いただく必要があります。。

Android アプリの使い方

この項目では、AI Chip 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。 ※Android アプリにより AI Chip と通信を行う際は、Bluetooth モジュールをご購入いただく必要があります。

2. Al Chip の使い方

2.1 AI Chip の入出力装置

1. CPU 用電源スイッチ (図.1-1)



図.1-1: CPU 用電源スイッチ(前面から)





図.1-2:モータ用電源スイッチ(背面)

3. プッシュスイッチ×2 (図.1-3)(図.1-4)



図.1-3: プッシュスイッチ場所(表面から)



図.1-4: プッシュスイッチ場所(後部)

2.2 AI Chip の充電方法

本製品は付属の充電器にバッテリを接続後、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 が赤く発光している時はバッテリの充電中です。また充電器の LED が緑色の時はバッテリの充電が完了しています。 なお、充電の際ケーブルを持ってコネクタの取り付け取り外しをすると、ケーブルなどの部品が破損する恐れがあります。 怪我や事故の原因となりますので、充電の際は必ずコネクタ部分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張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。

2.3 AI Chip の起動方法

はじめに AI Chip の配線が図の通りかを、確認をしてください。(図.1-5)





図.1-5: 通常の配線

図.1-6: 誤った配線

配線に関しては「AlChipV3 配線マニュアル」に詳しく記載されています。必ず熟読の上、配線作業を行ってください。

機体を図.1-5 の向き置いた時に図の上側にバッテリのケーブルが接続されていて、下側に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。

電源の土を誤って配線をすると基盤が故障する為、必ず逆配線をしないでください。

また、図.1-6 のように電源とモータを間違えて配線しても基盤が故障する為、逆配線しないよう注意が必要です。

Al Chip には電源スイッチが二箇所あります。

一つが機体上部にある CPU 用電源スイッチです。

こちらは LiPo バッテリの正面にスライドスイッチがあるため、それを「ON」または「+」側へスライドする事で電源が入ります。

もう一つは機体下部にあるモータ用電源スイッチです。こちらは機体の裏側についているため、起動時は「ON」の角度までスイッチを回 してください。

この二つの電源をつけることで Al Chip は起動します。

2.4 AI Chip の起動モード

Al Chip 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モードを切り替えることが出来ます。 切り替えは機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の通りとなります。



プッシュスイッチ を押す回数	緑 LED 発光パターン	モード名称	モードの内容
0	消灯	何もなし	なし
1	点灯	前進モード (duty 最大)	右のプッシュスイッチを押すとモータ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。
2	0.5 秒間隔点滅	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の速度を変動させます。 このモードでは Al Chip は前後進します。
3	0.25 秒間隔点滅	自律走行モード	右のプッシュスイッチを押すことで Al Chip 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。
4	0.125 秒間隔点滅	Bluetooth setting モード	Buluetooth の通信設定などの変更が行えるモードです。
5	0.062 秒間隔点滅	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。

常に Al Chip からデータが送信され、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場合は、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場合、このモードで起動します。

3. Processing サンプルプログラムの使い方

3.1 Processing の導入

Processing のダウンロードページ(https://processing.org/download/)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。(図.2-1)



図.2-1: ダウンロード画面1

ダウンロード画面へ遷移すると、画面の一番上に最新のバージョンのダウンロード項目が表示されます。最新版を利用する場合には、この 項目のうち、お使いの OS が表記されているのリンクを選択してください。(図.2-3)



図.2-3: ダウンロード画面 2

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。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。

3.2 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 AlChipV3 のページ(https://github.com/rt-net/AlchipV3)からダウンロードしてください。 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。(図.2-4)

Search or jump to	Pull requests Issues Marketplace	e Explore	\$ +· 1·
Ê rt-net / AlchipV3		O Unwatch	• 1 🛣 Star 0 Fork 0
↔ Code ① Issues ② ① Pull request	ts 🛈 🔢 Projects 🛈 🔝 Wiki 🔄 Ins	sights 🔅 Settings	
No description, website, or topics provided. Add topics			Edit
① 12 commits		0 releases	1 contributor
Branch: master New pull request	Ì	Create new file Upload file	s Find file Clone or download *
👬 kenjiinukai revise md file			Latest commit c84283b 18 days ago
Android/BloothoothChat	Initial commit		22 days ago
MCUXpresso	プログラムアップデート		22 days ago
Processing	Initial commit		22 days ago
Circuit	回路について追記		22 days ago
🖿 firmware	V3用プログラム&ファームフェアU	P	22 days ago
in manual	Add cable manual		18 days ago
README.md	revise md file		18 days ago
I README.md			

図.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。

3.3 サンプルプログラムの実行

Processing の起動

まず[3.1 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、[3.2 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」を開きます。

パソコンと AI Chip の Bluetooth のペアリングと通信ポートの設定

開いたら、パソコンと AI Chip 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。(図.2-5)



Windows パソコンの場合

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

 ← → ◆ ↑ 回, 32/k0-k/(RA) → ✓ 8 2)/ビューターの設定を調整します ジステムとせきユリティ 2)/ビューターの設定を調整します ジステムとせきユリティ 2)/ビューターの設定を調整します ジステムとせきユリティ 2)/ビューターの設定を調整します ジステムとせきユリティ 2)/ビューターの設定を調整します ジステムとせきユリティ 2)/ビューターの設定を調整します 2)/ビューターの設定を調整します ジステムとせきユリティ 2)/ビューターの設定を調整します ジステムとせきユリティ 2)/ビューターの設定を調整します アンクリーク アカウント ジスクトンドの登場の変更 デスクトッブのカスタマイズ アスクトップのカスタマイズ やられ、雪茄、および地域

図.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 Al Chip 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。



▋ デバイスを追加します			-		×
パスコード を比	べてください				
	RNBT-672Cのパスコードは、これと一致し 092508	ていますか?			
	[(\$U(Y)	いいえ(<u>N</u>)	キャン	セル

図.2-7: デバイスの選択画面

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。(図.2-7)

図.2-8: ペアリングコード比較画面

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。(図.2-8)

このデバイスは、このコンピューターに正常に追加されました Windows で現在ドライバーを確認中であり、必要に応じてそれらを インストールします。デバイズが使用できるようになる前に、この	•
処理の売了を特たなければならない場合があります。 このデバイスのインストールが正しく売了したことを確認するに は、[<u>アバイスとブリンター</u>] でこのデバイスを参照してください。	RT-AICHIP-07



図.2-9: 接続完了画面

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。(図.2-9)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。 図.2-10: Bluetooth アイコンの選択

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。

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。(図.2-10)



Fig.2-11: Bluetooth の設定画面

「設定を開く」をクリックすると、「Bluetooth とその他の デバイス」の画面が開かれます。開かれたら、右側の「その 他の Bluetooth オプション」を選択したください。(Fig.2-11)

Fig.2-12: COM ポート確認画面

「その他の Bluetooth オプション」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」を選択してください。「COM ポート」を選択する と、パソコンに接続された Bluetooth 機器の一覧が表示されます。 この一覧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 ポートの名称を確認してください。(Fig.2-12)

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。

Macintosh パソコンの場合

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



図.2-13: システム環境設定画面

図.2-14: Bluetooth 接続画面



図.2-15: Bluetooth 接続確認

	Bluetooth	٩	0
	- 112		1
	RNBT-4EA9 接続済み		0
	\sim		
Bluetooth : 入			
Bluetooth を切にする			
以下の名前で検出可能:			
"Nakagawa Ø MacBook Pro"			
	X=====K=K Bluetooth	を表示 詳細	11日本 2
		a were	

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)

接続が完了すると、「接続済み」の表記になります。(Fig.2-15)

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



図.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-××××-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後、processing のプログラムを起動させてください。(図.2-17)



図.2-17: processing の実行ポタン

起動するとウィンドウが立ち上がります。ウィンドウ上でマウスクリックをすると、Al Chip の緑の LED が点灯し、エディタ下部にはログ 出力が表示されます。ここには Al Chip からのデータが出力されます。 データの詳細については以下をご覧ください。(Table.2-1)(Table.2-2) AI ChipV3 マニュアル

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) 23

Table.2-1

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

Table.2-2

3.4 サンプルプログラムの解説

サンプルプログラムの機能

プログラムを実行すると、ウィンドウが開きます。そのウィンドウ上をクリックすると、AI Chip の緑の LED が点灯します。 エディタ下部の、ターミナルに AI Chip 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 AI Chip のモータが動きます。

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

はじめに下記のプログラムが実行されます。

//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。

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 関数はプログラム開始直後、初めに一回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ください。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ご確認ください。

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色を変更しています。

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

serialEvent 関数は Alchip 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。

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、生成されたコマンドは port.write 関数を用いて AI Chip 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 <u>duty を-1.0 から 1.0 で指定</u> <br>
              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 & 0x00000ff;
 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;
}
```

AI ChipV3 マニュアル

このコマンドは keyPressed 内で実行されます。 引数の duty によって Al Chip のモータ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-
 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、AI Chip 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、Al Chip に送信しています。

コマンドの生成方法

Al Chip 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、Al Chip への命令の種類を変えることができます。 5byte 目以降はデータフェールドです。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。



101.2-1: コマント送信ノロトコルの説明

•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 の符号付整数になります。

上位 bit が 5byte 目、下位 bit が 6byte 目な為、配列の順番に気をつけてください。



ref.2-2: モータ制御コマンド

○モータの 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。

• 4	LEC ज(緑)L) の EDの制	点灯 ^{御コマ}	制御 マンド	₽⊐⋜	マント	*					
	Obyte	1byte	2byte	3byte	4byte	5byte	6byte	7byte	8byte	9byte		
	h	eadde	r	id			data	field				
	99	109	100	1	XX	0	0	0	0	0		
۰ź	E(赤)L	ED の 制	御コマ	マンド	1:LED	点灯 0:	LED消火	Ţ				
	Obyte	1byte	2byte	3byte	4byte	5byte	6byte	7byte	8byte	9byte		
	h	eadde	r	id			data	field				
	99	109	100	2	XX	0	0	0	0	0		
					1:LED	点灯 0:	LED消火	Ţ			16	

ref.2-3: LED 制御コマンド

4. Android アプリの使い方

4.1 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.パッケージファイルから直接インストールする方法

ダウンロードした「AlchipV3」フォルダから「Android」→「BloothoothChat」→「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「Applicationdebug.apk」を直接起動することでインストールできます。

2.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

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た後、「DOWNLOAD OPTION」に各 OS ごとに用意されたインストールパッケージをダ ウンロードしてください。(図.3-1)

A	Andı	roid Studio downle	bads	5
	Platform	Android Studio package	Size	SHA-256 checksum
	Windows	andreid-studio-ide-173.4819257-windows.exe Recommended	758 MB	2d11cd16ffefc7f4aca82bd95b0d0ca849a854a07ba1a353adf65dfc102aee9b
((64-bit)	android-studio-ide-173.4819257-windows.zip No .exe installer	855 MB	6941761a9324998d9cdf5d7548ff16fcb65c9e71ea70e1bc75d066b51b77c7dd
1	Windows (32-bit)	android-studio-ide-173.4819257-windows32.zip No .exe installer	854 MB	85cbfb33c94183abcb70caac34ea034214a079376fb1b2e10bc7b4ed71c05cf0
,	Mac	android-studio-ide-173.4819257-mac.dmg	849 MB	d4a8502c5aabfc5477ff30dfffe296bf705bd7e62650a76796b646a8f28b5e5c
	Linux	android-studio-ide-173.4819257-linux zip	853 MB	d86748e44d658fd39581b40f7b706fb397fc1eca5dd6f8066a56c0beb856dea4
s	ee the And	Iroid Studio release notes.		

図.3-1: Android Stuido のダウンロードページ

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 AI Chip V3 の github のページより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」を選択してください。(図.3-)2

O Instruction	Full requests insues Machetplace Explore	\$ +· 11-
1 m-net / AlchipV3		Olimath* 1 the 1 fet 1
O Cole () hive # 1) Put repo	mill (Project B. 10.996 (c) height (c)	Setings
Ko alexistiption, website, or tapics provide All have	ul.	line -
@Warments	jó 2 teanthra 🖒 8 millionn	41 Econolisius
Branch marker * New part request	Create new Ne	In Sphere Net First Star
C berginskal mine out the		Latent commit addditer 16 days age
In Andread Development Chat	Instal convert	23 stays ager
B MCD press	プログラムアップデート	22 ships again
In Processing	Initial converts	22 shiye age
State of the second s	開発について追認	22 days apr
Ba homourn	マリ用プログラムルファームフェアルF	22 days age
Brand	Add cable manual	10 daya agin
E READIVE ind	retries and file	Till shiye agar
BREADWE.out		

図.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ます。



解凍後「Android Studio」を起動し、左図起動画面の「Open an existing Android Studio project」を選択してください。(図.3-3)

図.3-3: Android Studio の起動画面

「Open an existing Android Studio project」で起動後、先ほど解凍した「Al Chip V3」のディレクトリ内にある「BluetoothChat」を選択 して開きます。このプロジェクトが、Al Chip 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 デバックモード」を ON にして PC と接続してください。

接続後、Android Studio 上部にある「Run」の中から緑の三角アイコン「Run」をクリックします。

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。(図.3-4)

Eile	: Edit Yiew Navigate Code Analyze Befactor Build	Run Tools VCS Window Help	
	I C: Mit Users Mit kenji 🕅 AppData 🖿 Local 🖿 And		
병			🕽 View.java 🗶 🛗 fragment_bluetooth_chat.xml 🗵 💿 Al_CHIP.java 🗵
Proj	🔻 🔚 BluetoothChat Ca¥Users¥kenji¥Documents¥Alchip-me		
R	Google Jaradie		7
12	▶ D∎ idea		
dar	Application		
Sta	build		
171	aradle		
w	Image: Screenshots	Attach to Local Process	
12	Not Street Stree	Edit Configurations	
pte	🙆 build.gradle	1 Import Test Results	
P.C	CONTRIB.md		
	CONTRIBUTING.md		
	🗿 gradlew		
	📋 gradlew.bat		view.ViewGroup) subclass is the base class for layouts , which
	🔮 LICENSE		ntainers that hold other Views (or other ViewGroups) and define
	🚮 local.properties		
	NOTICE		
	README.md		
	IIII External Libraries		on about using this class to develop your application's user interface.

図.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。 (図.3-5)

「OK」を選択して、書き込みを開始してください。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。

図.3-5: プログラム転送2

4.2 Android サンプルアプリの使い方



図.3-6: Android アプリ画面

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 AI Chip と情報の送受信が可能です。(図.3-6) AI Chip と通信をするためには、画面右上の Bluetooth アイコンから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ボタン「BACK」「Start」「STOP」を押すと Al Chip のモータが動

回回千人に能置された生」フジホスク、DACK」「Start」、Stort」を計算とArchipのピークが勤 きます。

右側二つのボタン「LED1」「LED2」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 AI Chip 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 AI Chip の加速度を示します。

Battery(V)の部分にはリアルタイムで AI Chip のモータ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 Al Chip 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、Al Chip が動きます。(Table.3-2)

音声入力キーワード				
音声入力	Al Chip 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.3-1 音声入力で反応するキーワード				

テキスト入力キ	テキスト入力キーワード				
テキスト入力	Al Chip 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				
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				
Table.3-2 テキスト入力で反応するキーワード					

4.3 Android アプリに付属しているライブラリの使い方

「BluetoothChatService.java」を継承した「AI CHIP.java」が AI Chip のライブラリです。

Al Chip からのデータの get 関数と Al Chip への set 関数が用意されていますので、Android アプリ内で関数を呼び出してご利用ください。 詳しくは「Al CHIP.java」をご参照ください。

改版履歴

発行日	ページ	改訂内容
2018/7/17	-	新規発行

AI ChipV3 マニュアル

製造元

株式会社アールティ 〒101-0021 東京都千代田区外神田 3-2-13 山口ビル 3F TEL 03-6666-2566 URL https://www.rt-net.jp/

製品に関するお問い合わせ

本製品に関するお問い合わせは、下記までお願いします。 お問い合わせは電子メールにて受け付けております。 E-mail: support@rt-net.jp