「JavaScript SDK for V-Sido CONNECT」 利用の手引き

—Intel Edison編—







目 次

1. はじめに

1-1. 本マニュアルの概要 p.	3
1-2. 使用する機材など ―――― p.	4
1-3. 各機器の接続イメージ ―――― p.	5
1-4. 開発環境の概略 ——————— p.	6
2. Intel Edisonを利用する手順	
2-1. Intel Edisonとパソコンを接続 ——— p.8	3
2-2. Intel Edisonの初期設定 ———— p.9)
2-3. Intel Edisonへのシリアル接続 ――― p.1	0
2-4. VSidoConn4Edisonのインストール ― p.1	1
2-5. Intel EdisonのIPアドレスを確認する ―― p.1	2
2-6. Bluetoothのペアリング ――― p.13	3
2-7. サンプルプログラムを参照する―――― p.14	4
2-8. サンプルプログラムを動かす―――― p.1!	5

3. JavaScriptによるロボット開発入門

3-1. ロボットの制御について ―――― p.1	7
3-2. HTML&JavaScriptの作成手順 ――― p.1	8
3-3. 「目標角制御」のサンプルコード ――― p.1	9
3-4. 「逆運動の利用」のサンプルコード ―― p.2	0
3-5. 「ロボットの歩行」のサンプルコード — p.2	1
3-6. ロボットの操作手順 ―――― p.2	2
3-7. 補足事項 ————— p.2	3
奥付	



1. はじめに

Asratec Corp, All rights reserved.

2



1-1. 本マニュアルの概要

このマニュアルでは「Intel Edison」を使って、「V-Sido CONNECT RC」を接続したロボットを制御する手順を示します。

このマニュアルの手順により、「JaveScript SDK for V-Sido CONNECT」がIntel Edisonで利用でき、JavaScriptによるロボット制 御が可能になります。Java Script SDK for V-Sido CONNECTの詳細は、 V-Sido開発者支援サイト「V-Sido Developer」で公開しているドキュ メントをご確認ください(https://v-sido-developer.com/)。





Intel Edison (Breakout Board Kit)



V-Sido CONNECT RC Bluetoothアダプタ実装済



ロボット GR-001/DARWIN-MINI

【その他に必要な機材、ソフトウェア】

- 開発用パソコン (Windows、Mac OSなど)
- 無線LANアクセスポイント
- Micro USBケーブル×2
- Intel Edisonのファームウェア(入手方法は後述)
- シリアル接続ターミナルソフト(「Tera Term」など)
- テキストエディタ(「メモ帳」など)
- Intel Edison用プログラム「VSidoConn4Edison」(入手方法は後述)



1-3. 各機器の接続イメージ





1-4. JavaScriptによる開発環境の概略





2. Intel Edisonを利用する手順



2-1. Intel Edisonとパソコンを接続

Micro USBケーブル×2を使って、Intel Edisonを開発用パソコンに接続します。





2-2. Intel Edisonの初期設定

Intel Edisonのファームウェアをアップデートしてください。アップデート方法など 詳細は、インテルの公式サイトなどを参照してください。

このマニュアルでの検証は、インテルの下記サイトで公開されている「Installers」 アプリケーションを利用して、Intel Edisonのファームウェアアップデートや無線 LANの初期設定などを行っています(Intel Edisonのバージョンは201606061707)。

https://software.intel.com/en-us/iot/hardware/edison/downloads





2-3. Intel Edisonへのシリアル接続

Intel Edisonのファームウェアのバージョンアップを行ったあと、ロボットを操作す るプログラム(VSidoConn4Edison)をインストールするため、パソコンからIntel Edisonにシリアル接続します。ここでは、フリーソフトウェアの「Tera Term」を 使った手順を解説します(https://osdn.jp/projects/ttssh2/)。

パソコンでTera Termを起動し、下図のように設定し、接続を行います。



「シリアル」で「ポート」にIntel Edison が接続されているポート(USB Serial Port)を選択し、「OK」ボタンを押す



次に、メニューで[設定] – [シリアルポー ト]を選び、「ボー・レート」を115200に 設定して、「OK」ボタンを押す

2-4. VSidoConn4Edisonのインストール



シリアル接続後、Tera Termで「Enter」キーを押すと、ログインが促されるので、 「**root**」ユーザーでログインします。

もし、Intel Edisonのネットワーク設定がまだの場合、下記のコマンドを実行して WiFiの設定を行ってください。

configure_edison --wifi

Intel Edisonがインターネットに接続された状態で、「VSidoConn4Edison」のイン ストールを行います。このプログラムはアスラテックが提供しており、下記のコマン ドでインストールが行われます。

wget --no-check-certificate -O - https://asratec.github.io/VSidoConn4Edison/install.sh | sh

インストール後、下記のコマンドを実行してIntel Edisonをシャットダウンします。

poweroff

"[OK] Reached target Shutdown."のメッセージが表示されたあと、USBケーブルの抜き差しを行って、Intel Edisonの電源を入れ直します。



2-5. Intel EdisonのIPアドレスを確認する

Intel Edisonを再起動させたあと、再び「Tera Term」でIntel Edisonにシリアル接続して、rootでログインします。ログイン後、Intel EdisonのIPアドレスを確認するために、下記のコマンドを実行します。

ifconfig wlan0

コマンドの実行結果で「inet addr」の項目に表示されているアドレスが、Intel EdisonのIPアドレスになります。

COM3:115200baud - Tera Term VT	- • ×
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)	
root@edison:~# ifconfig wlan0 wlan0 Link encap:Ethernet HWaddr 78:4b:87:a1:03:e1 inet addr:192.168.11.8 Deast:0.0.0.0 Mask:255.255.255.0 UP DROADCAST RUNNING MULTICAST MTU:1500 Metric:1 RX packets:31 errors:0 dropped:0 overruns:0 frame:0 TX packets:84 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:3000 (2.9 KiB) TX bytes:8696 (8.4 KiB) root@edison:~#	この例では、IPアドレスは 192.168.11.8となります。 本マニュアルでは以降、こ のIPアドレスで説明します が、適宜、各ユーザーの環 境のIPアドレスに読み替え てください



2-6. Bluetoothのペアリング

次に、Intel EdisonとV-Sido CONNECT RCとのBluetoothペアリングを行います。 V-Sido CONNECT RC (Bluetoothアダプタセットを搭載済みのもの)をロボットに 接続して、ロボットの電源をONにします。

開発用パソコンのWebブラウザで「**http://192.168.11.8:8089**」を開き、Web ページを「Robot Control」→「ロボット検出」→「スキャン」と順に文字をクリッ クします。

名称	MACアドレス	接続
SBDBT-001bdc04c3e8	00:1B:DC:04:C3:E8	接続

Bluetoothデバイスが検出されるので、適切なデバイスの箇所にある「接続」ボタン を押して、Bluetoothペアリングを実行します。「接続」ボタンを押して30秒ほど 経ってから、USBケーブルを抜き差ししてIntel Edisonの電源を入れ直してください。

以上で、Intel EdisonとV-Sido CONNECT RCを連係させる準備は完了です。



2-7. サンプルプログラムを参照する

Intel EdisonにインストールしたVSidoConn4Edisonには、サンプルプログラムが用意されています。下記のURLでアクセスできるので、開発用パソコンのWebブラウザでアクセスして、ロボットの動作を確認してください。

http://192.168.11.8:8089

なお、IPアドレスは、適宜、ご自身のIntel EdisonのIPアドレスに読み替えてください。

このサンプルプログラムは、 JavaScript SDK for V-Sido CONNECTで作られています。実体は、Intel Edisonの以下のディレクトリに格納されているので、ソースコードを参照するなどして、開発の参考にしてください。

/home/sysroot/usr/share/WebServer/



2-8. サンプルプログラムを動かす

「http://192.168.11.8:8089」のWebページにある「Robot Control」の文字をク リックすると、サンプルの一覧が表示されます。これらの機能の多くは、V-Sido CONNECT RCのコマンドに対応しています(別マニュアル『V-Sido CONNECT RC コマンドリファレンス』参照)。

「/home/sysroot/usr/share/WebServer/sample」フォルダ下にあるソースコードと、実際のロボットの動きを確認して、開発の参考にしてください。

V-SidoWeb TargetAngel x	1	-		×
← → C 🗋 192.168.110.31:8089/sample/target2.html			ŝ	≡
IPアドレス 歌定				
ホームに戻る				
トグル選択				
角度取得				
角度設定				
ServolD:1 0				-
ServoID.2 30				
ServoID:3 0				
ServoID:4 0				÷

「目標角度設定」の動作を確認できるWebページ。任意のサーボIDを指定して数値を入力したあと、「角度設定」をクリックすることで、指定したサーボが指定した角度に変わります。

なお「IPアドレス」の箇所は同一 ネットワーク内の別のロボットを制 御する場合に指定する項目なので、 通常は無視して構いません



3. JavaScriptによるロボット開発入門



3-1. ロボットの制御について

JavaScript SDK for V-Sido CONNECTを使ったロボットの制御には次の2つを使用します。





3-2. HTML&JavaScriptの作成手順

- ① 開発用パソコン上でテキストエディタを用い、任意の.html拡張子のファイルを 作成します(たとえばindex.htmlなど)。
- ② ロボットを制御するためのコードを記述します。サンプルコードとして、次ページ以降に3種類の動作を用意しましたので、このいずれかの内容をコピーして、 先ほど①で作成したファイルにペーストしてください。
 - 3-3.「目標角制御」のサンプルコード
 - 3-4. 「逆運動の利用」のサンプルコード
 - 3-5. 「ロボットの歩行」のサンプルコード
- ③ ②のサンプルコードに記述されているIPアドレスを、適切なIPアドレス(ご利用の環境でのIntel EdisonのIPアドレス。本ドキュメントの2-7で確認したもの)に変更し、UTF-8形式で保存します。



3-3.「目標角制御」のサンプルコード

```
<!DOCTYPE html>
<html>
<head>
     <meta content="text/html;charset=shift jis">
    <title>指定したサーボを動かす</title>
    <!- 必須:ロボットの制御をするためのライブラリ - 適切なIPに編集してください -->
     <script src="http://192.168.11.8:8088/ip2/js/vsido.client.api.js"></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></sc
     <script>
                var connect = new VSidoWeb({ "ip":"192.168.11.8"}); /* 適切なIPに編集してください */
                function exec(){
                                                                                                                                                                              /* exec関数 */
                                                                                                                                                                        /* 目標関節角度コマンドの生成 */
                           var angle = new vsido.SetServoAngle(100);
                                                                                                                                                                             /* 目標関節角度に到達までの時間 100 ミリ秒 */
                                                                                                                                                                             /* サーボIDの指定(2),目標関節角度の指定(60)*/
                           angle.addAngle(2,60)
                                                                                                                                                                              /* コマンドの実行 */
                           connect.send(angle);
                 1
     </script>
</head>
<body>
    <button type="button" onclick="exec();">実行</button> /* ボタンクリックでexec関数を実行 */
</body>
</html>
 ロボットによってサーボIDが示す関節が異なる点にご注意ください。たとえばサーボ
```

ID2は、GR-001では頭部を示し、DARWIN-MINIでは左肩部分を示します



3-4. 「逆運動の利用」のサンプルコード

<!DOCTYPE html> <html> <head> <meta content="text/html;charset=shift jis"> <title>右手を手前に出す</title> <!- 必須:ロボットの制御をするためのライブラリ - 適切なIPに編集してください --> <script src="http://192.168.11.8:8088/ip2/js/vsido.client.api.js"></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></sc <script> var connect = new vsido.Connect({"ip":"192.168.11.8"}); /*適切なIPに編集してください */ /* exec**関数** */ function exec() { var ikflag= new vsido.SetIK(ikflag); ikflag["position"] = true; /* 位置 */ /* 姿勢 */ ikflag["rotation"] = false; ikflag["torque"]= false; /* **トルク** */ /* IK**コマンドの生成** */ var ik = new vsido.SetIK(ikflag); /* IKコマンドパラメータの生成 */ var kdt = vsido.kdt(); ik.addPostion(2,0,-100,0); /* 制御するIK部位の指定(右手:2) */ /* x軸での可動範囲の位置割合 0 */ /* v軸での可動範囲の位置割合 -100*/ /* z 軸での可動範囲の位置割合 0*/ /* コマンドの実行 */ connect.send(ik); } </script> </head> <body> /* ボタンクリックでexec 関数を実行 */ </bodv> </html>



3-5. 「ロボットの歩行」のサンプルコード

```
<!DOCTYPE html>
<html>
<head>
<meta content="text/html;charset=shift_jis">
<title>口ボットを前へ歩かせる</title>
```

<!- 必須:ロボットの制御をするためのライブラリ - 適切なIPに編集してください -->

```
<script src="http://192.168.11.8:8088/ip2/js/vsido.client.api.js"></script>
<script>
```

```
var connect= new vsido.Connect ({ "ip":"192.168.11.8"}); /* 適切なIPを指定 */
```

```
function exec() { /* exec関数 */
var motion = vsido.walk(50,0); /* 歩行コマンドの生成 */
/* 歩行可能最高速度の50%で前進させる */
/* 回転成分 0% */
connect.send(motion); /* コマンドの実行 */
}
</script>
</head>
<body>
<button type="button" onclick="exec();">実行</button> /* ボタングリックでexec関数を実行 */
</body>
</html>
```



3-6. ロボットの操作手順

- ロボットの電源をONにして、Intel EdisonとV-Sido CONNECT RCとの Bluetoothペアリングが完了するまで待ちます(SBDBTのLED2が点灯から点 滅に変わります)。
- 3-2で作成したファイルをダブルクリックして、Webブラウザで開きます (ダブルクリックで開かなかった場合、直接Webブラウザへドラッグ&ド ロップします)。
- ③ Webブラウザ上でUIを操作し、ロボットの操作を行います。



3-7. 補足事項

- ✓ JavaScript SDK for V-Sido CONNECTのAPIは、 V-Sido開発者支援サイト 「V-Sido Developer」で公開していますので、そちらを参照してください (<u>https://v-sido-developer.com/learning/reference/v-sido-connect-web-api-js/</u>)
- ✓ V-Sido Developerでは、そのほかV-Sido CONNECT RCの開発者向けの情報 を公開しています。フォーラムもありますので、不明点などがありましたら、 そちらもご確認ください。





• 記載された社名、製品名は一般に各社の商標または登録商標です。

「JavaScript SDK for V-Sido CONNECT」利用の手引き――Intel Edison編 Ver. 1.00

> アスラテック株式会社 〒101-0042 東京都千代田区東松下町45

> > 公開日:160810