

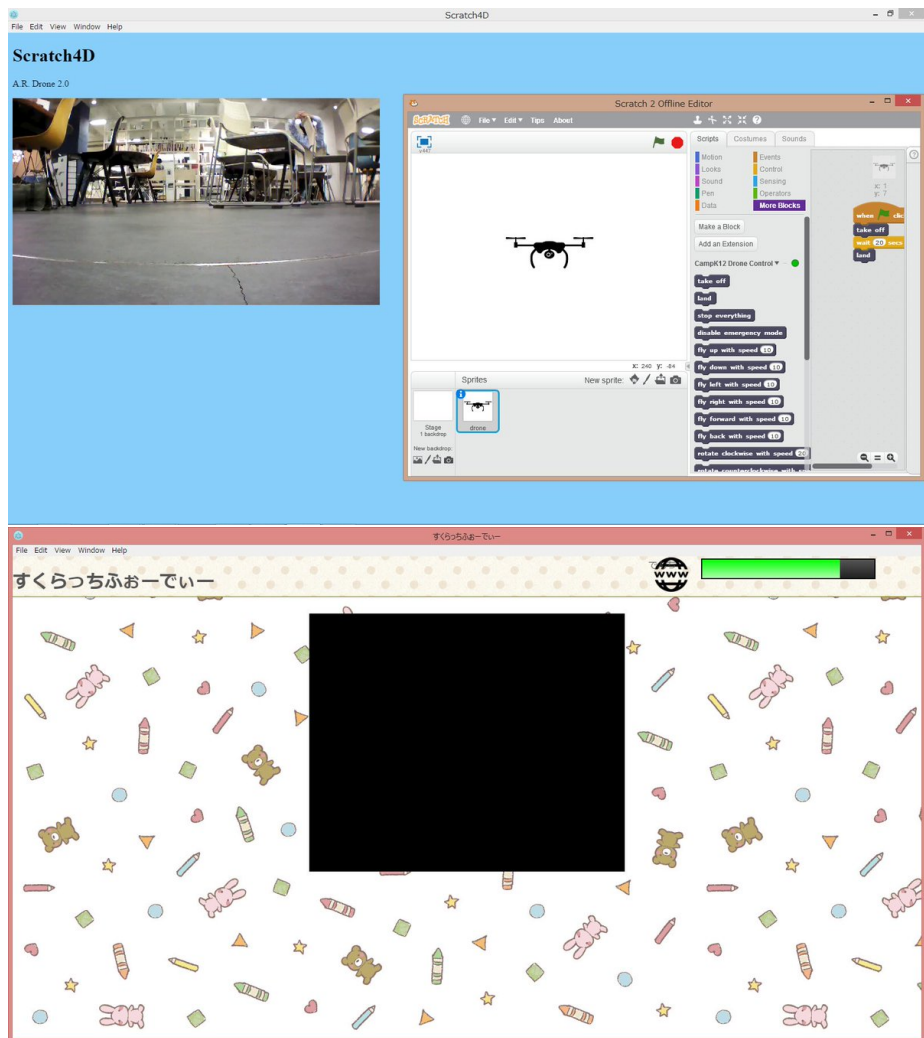
## Contents

1	Scratch4D	1
2	Scratch4Dの使い方	2
2.1	Windows	2
2.2	Mac	3
2.3	Linux	3
2.4	ARM Linux	4
3	Scratch4Dの仕組み	4
3.1	初心者プログラマでも簡単にドローンプログラミングができるための工夫	4
3.2	Scratch Extensions	5
3.2.1	Scratch拡張機能の仕組み	5
3.2.2	Scratch拡張機能を実現する方法	5
3.2.3	拡張プラグインを作る	5
3.2.4	Scratch Helper Appを作る	6
4	ElectronでHelper Appを実装する	6
4.1	Electronとは	6
4.2	Electronの仕組み	8
4.3	Node.jsとは	8
4.4	サンプル：Node.jsでHTTPサーバーを作る	8
4.5	Node.jsの仕組み	9

## 1 Scratch4D

Scratch4DはScratch HTTP Extensions & electronを使ってscratchでドローンを操縦できるようにしたソフトウェアです。実行ファイルをダブルクリックするだけで起動できます。

他にも、Scratchでは出来ないドローンの動画ビューの確認や、離着陸ボタンなどでドローン进行操作することもできます。



## 2 Scratch4Dの使い方

### 2.1 Windows

1. Scratch4Dをダウンロードして、実行ファイルをクリックして起動します。  
<https://drive.google.com/file/d/0By3Sz9jL9bM3Z1l6eE1UbkhTdQ/view?usp=sharing>
2. Scratch4Dを解凍して、フォルダ内のScratch4D.exeをダブルクリックして実行してください。

## 2.2 Mac

1. Scratch4Dのgithubのレポジトリをクローンしてください。

```
git clone https://github.com/kendemu/Scratch4D
```

2. NVMをインストールしてください。

```
wget -qO- https://raw.githubusercontent.com/creationix/nvm/v0.33.0/install.sh | bash
```

3. NVMでNode.jsのバージョンをv6.9.0 LTSにしてください。

```
nvm use v6.9.0
```

4. 動画エンコーディングライブラリffmpegをインストールしてください。

```
brew install ffmpeg
```

5. Scratch4Dのディレクトリ下でelectronプロジェクトをビルドしてください。

```
npm -g i asar  
npm run build
```

6. Scratch4Dのディレクトリ下でelectronプロジェクトをMacプラットフォーム用にデプロイしてください。

```
npm -g i electron-packager  
npm run deploy:mac
```

7. Scratch4D-(プラットフォーム名)-(CPUアーキテクチャ名)という名前のディレクトリが生成されます。  
そのディレクトリ下でプログラムを実行してください。

## 2.3 Linux

1. Scratch4Dのgithubのレポジトリをクローンしてください。

```
git clone https://github.com/kendemu/Scratch4D
```

2. NVMをインストールしてください。

```
wget -qO- https://raw.githubusercontent.com/creationix/nvm/v0.33.0/install.sh | bash
```

3. NVMでNode.jsのバージョンをv6.9.0 LTSにしてください。

```
nvm use v6.9.0
```

4. 動画エンコーディングライブラリffmpegをインストールしてください。

```
sudo apt-get install ffmpeg
```

5. Scratch4Dのディレクトリ下でelectronプロジェクトをビルドしてください。

```
npm -g i asar  
npm run build
```

6. Scratch4Dのディレクトリ下でelectronプロジェクトをLinuxプラットフォーム用にデプロイしてください。

```
npm -g i electron-packager
npm run deploy:linux
```

7. Scratch4D-(プラットフォーム名)-(CPUアーキテクチャ名)という名前のディレクトリが生成されます。  
そのディレクトリ下でプログラムを実行してください。

## 2.4 ARM Linux

1. Scratch4Dのgithubのレポジトリをクローンしてください。

2. NVMをインストールしてください。

```
wget -qO- https://raw.githubusercontent.com/creationix/nvm/v0.33.0/install.sh | bash
```

3. NVMでNode.jsのバージョンをv6.9.0 LTSにしてください。

```
nvm use v6.9.0
```

4. ffmpegをインストールしてください。ARM Linux用パッケージがaptレポジトリにないのでソースインストールしてください。  
<http://www.jeffreythompson.org/blog/2014/11/13/installing-ffmpeg-for-raspberry-pi/>

5. Scratch4Dのディレクトリ下でelectronプロジェクトをビルドしてください。

```
npm -g i asar
npm run build
```

6. Scratch4Dのディレクトリ下でelectronプロジェクトをARM Linuxプラットフォーム用にデプロイしてください。

```
npm -g i electron-packager
npm run deploy:arm
```

7. Scratch4D-(プラットフォーム名)-(CPUアーキテクチャ名)という名前のディレクトリが生成されます。  
そのディレクトリ下でプログラムを実行してください。

## 3 Scratch4Dの仕組み

Scratch4Dは「初心者プログラマでも簡単にドローンプログラミングができる」ことを目的として作られたソフトウェアです。

### 3.1 初心者プログラマでも簡単にドローンプログラミングができるための工夫

1. Windowsでのサポートに最も力を入れました。  
初心者プログラマは多くの場合Windowsを使っています。
2. アイコンのダブルクリックでソフトウェアを起動できるようにしました。  
初心者プログラマは、環境構築が嫌いです。

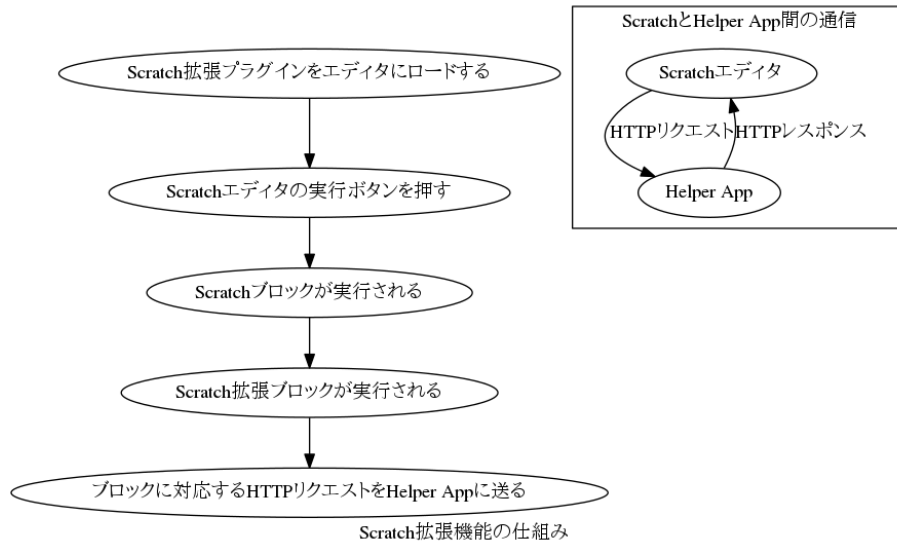
3. 動画ビューやドローンの操作がGUIでできるようにしました。  
初心者プログラマは、CUIやコマンドラインが嫌いです。GUIが大好きです。

## 3.2 Scratch Extensions

ScratchにはScratchXと呼ばれる、拡張機能があります。  
この拡張機能を使うことによって、  
ScratchでArduino, Kinect, Leap Motionなどのハードウェアと  
連携させることができますようになります。

### 3.2.1 Scratch拡張機能の仕組み

以下のようなフローでScratch4Dは拡張機能を実現しています。



### 3.2.2 Scratch拡張機能を実現する方法

拡張機能を実現するためには、2つのことを行う必要があります。

1. Scratchの拡張プラグインをJSON形式で作成する。
2. Scratch Helper Appと呼ばれるプログラムを作る。

### 3.2.3 拡張プラグインを作る

プラグインは拡張子が.s2eのJSON形式ファイルになります。  
プラグインの公式仕様書を読んで作ってください。

<https://wiki.scratch.mit.edu/w/images/ExtensionsDoc.HTTP-9-11.pdf>

サンプルとして、Arduino For Scratchのプラグインを貼ります。

```
{
  "extensionName": "A4S (Arduino For Scratch)",
  "extensionPort": 12345,
  "blockSpecs": [
    [" ", "set pin %n high", "pinHigh", 13],
    [" ", "set pin %n low", "pinLow", 13],
    [" ", "set pin %n as output", "pinOutput", 13],
    [" ", "set pin %n as input", "pinInput", 13],
    [" ", "set pin %n as %m.mode", "pinMode", 2, "Digital Input"],
    [" ", "digital write pin %n %m.highLow", "digitalWrite", 13, "high"],
    [" ", "analog write pin %n value %n", "analogWrite", 3, 255],
    [" ", "servo write pin %n degrees %n", "servoWrite", 5, 180],
    ["b", "digital read pin %n", "digitalRead", 2],
    ["r", "analog read pin %n", "analogRead", 0],
  ],
  "menus": {
    "mode": ["Digital Input", "Digital Output", "Analog Input", "Analog Output(PWM)", "Servo"],
    "highLow": ["high", "low"],
  },
}
```

### 3.2.4 Scratch Helper Appを作る

Scratch Helper Appは、Scratchとハードウェアのインターフェースの役目をします。

Scratch Editorから送信されたHTTPリクエストを元に、ハードウェアを操作します。

Scratch Helper Appは、HTTPリクエストを処理し、HTTPレスポンスを返すことができれば、

どのようなプログラミング言語でも実装可能です。

Javaで書かれていることが多いです。

(多分HTTPやGUIの実装がやすく、プログラミングしやすいため)

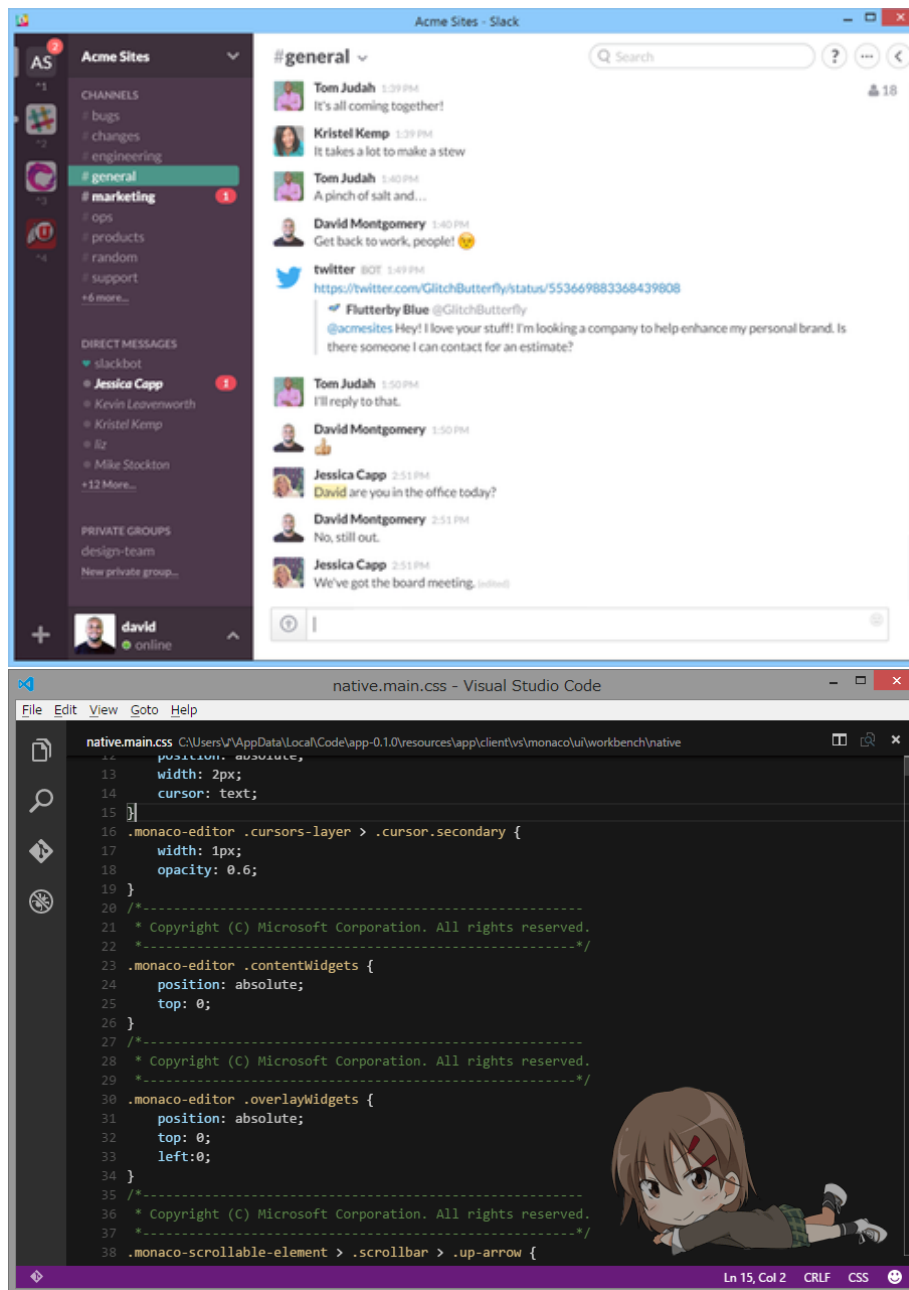
## 4 ElectronでHelper Appを実装する

今回はElectronでHelper Appを実装します。

### 4.1 Electronとは

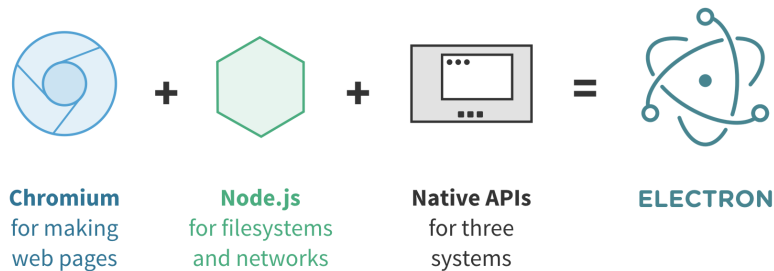
Electronはデスクトップアプリを開発するためのフレームワークです。

SlackのデスクトップアプリやMicrosoft Visual Studio CodeがElectronを使って作られています。



## 4.2 Electronの仕組み

ElectronはchromiumブラウザとNode.jsを合体させたフレームワークです。  
GUIはWEB技術のHTML5で設計でき、他のプログラムはNode.jsで作ります。



## 4.3 Node.jsとは

Node.jsはサーバーサイドでJavaScriptを実行できるようにした実行環境です。本来JavaScriptはWEBブラウザ上でしか動きませんでしたが、Node.jsを使うとJavaScriptでPythonやJavaと同じように汎用的なプログラムが作れます。コーディングされたJavaScriptがシングルスレッドで非同期に実行されることが特徴です。少ないコンピューター資源でサーバーが作れることで有名です。C++でコーディングされたJIT VMのGoogle V8 Engineを使用しているため、高速でJavaScriptを実行することができます。最近では組み込み機器とNode.jsを組み合わせるとIoTにも使われています。主にWEBサーバー開発やWEBフロントエンド開発に使用されます。

## 4.4 サンプル：Node.jsでHTTPサーバーを作る

```
const http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(8080);
console.log('Server running at http://127.0.0.1:8080/');
```



## 4.5 Node.jsの仕組み

Node.jsには3つの仕組みがあります。

1. 簡単に軽量なサーバープログラムが作れる
2. NPM/パッケージマネージャーを使って豊富なライブラリをすぐに使用することができる
3. npm scriptで開発工程が自動化・効率化できる