

# IoTハッカソン～business～ 事前勉強会【NefryでIoTハinzオン】Day 1

---

## 目次

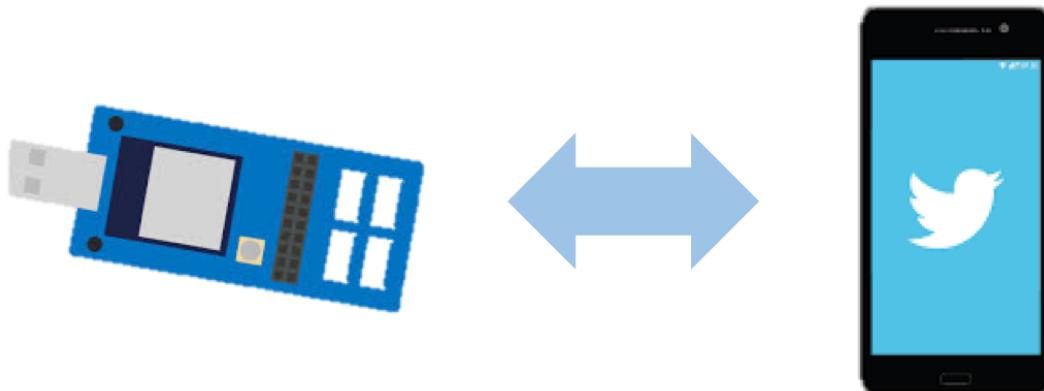
---

1. ノンプログラミングでTwitter投稿を行ってみよう
  2. LEDを光らせるプログラムを書いてみよう
  3. 光センサの情報を取得して記録してみよう
- 

## 1.NefryとIFTTTを用いて ノンプログラミングでTwitter投稿を行おう

---

<https://dotstud.io/docs/nefrybt-ifttt/>



---

Nefry BTとは？

IFTTT等のインターネットのサービスとハードウェアを簡単に接続できるIoTデバイス。Wi-Fi / BLE通信モジュール「ESP-WROOM-32」を搭載



電源にぶっ挿したまま、  
Wi-Fi経由でリモート開発。

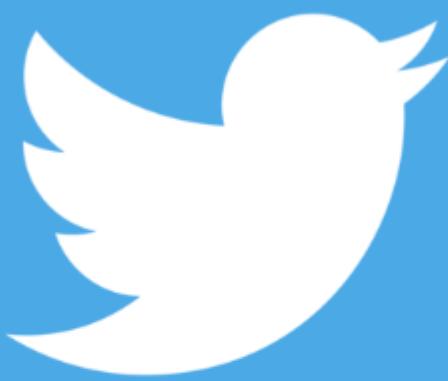
Nefru<sup>BT</sup>  
[ねふりーBT]

IFTTT (if this, then that) とは？

LINE, Twitter, Slack など、様々なWebサービス同士を 簡単に連携(レシピと呼ぶ)できるサービス



Instagram



Twitter



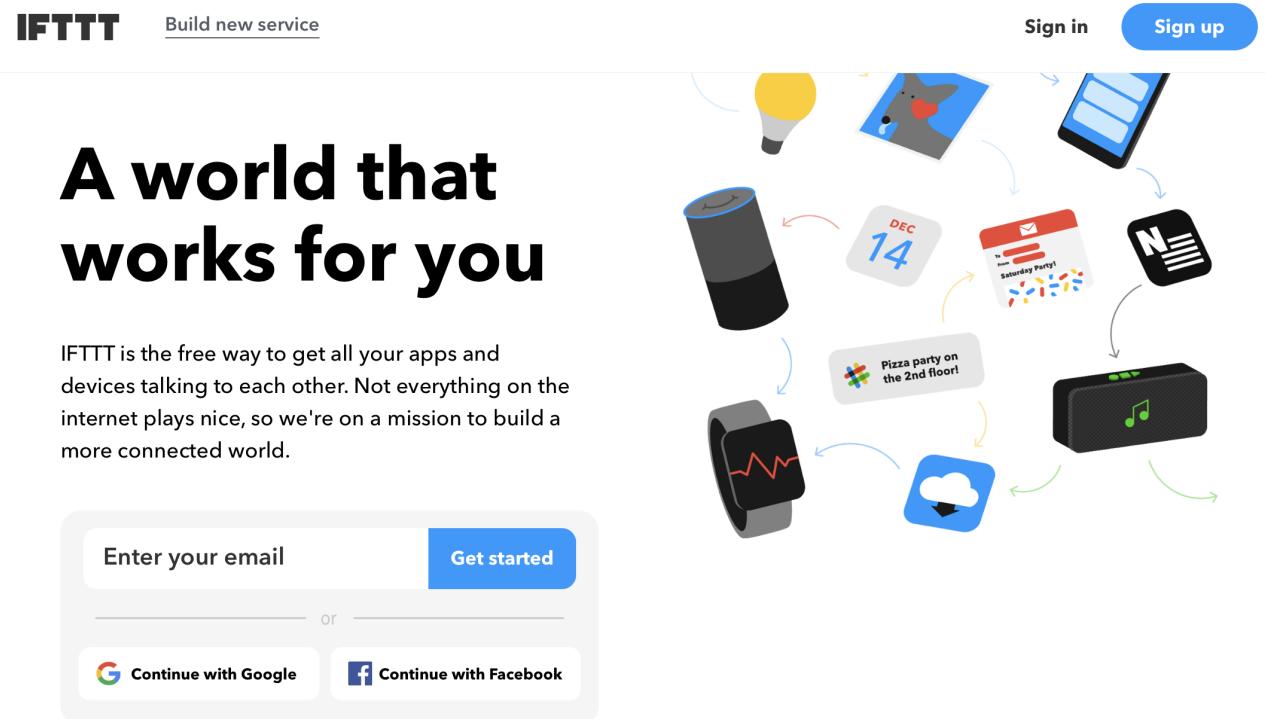


NefryとIFTTT連携の手順

- (1) IFTTTにレシピを作成 → (2) Secret Key を取得  
→ (3) Nefry BT の設定ページでSecret Key を登録

## (0) IFTTT,twitter のアカウント作成

- IFTTTのアカウントを作成 <https://ifttt.com>



- (アカウントを取得していない人のみ) twitter のアカウントを作成 <https://twitter.com/>




---

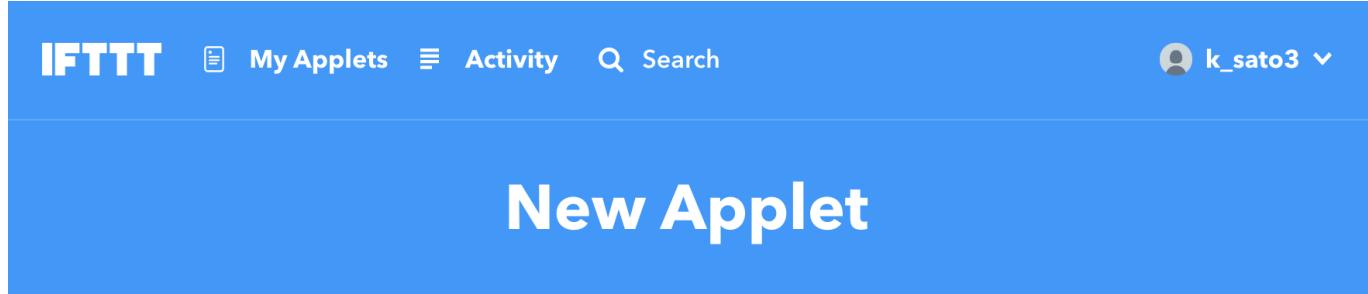
## (1) IFTTTにレシピを作成

レシピ： if this then that ○○ が起きたら △△ する Nefryのボタンを押し たら IFTTTのWebhooksにアクセスする

---

## (1) IFTTTにレシピを作成

[Thisの設定] 「+this」 をクリック



# if +this then that

Want to build your own service? Build on the platform [↗](#)

[About](#)   [Blog](#)   [Help](#)   [Jobs](#)   [Terms](#)   [Privacy](#)   [Trust](#)

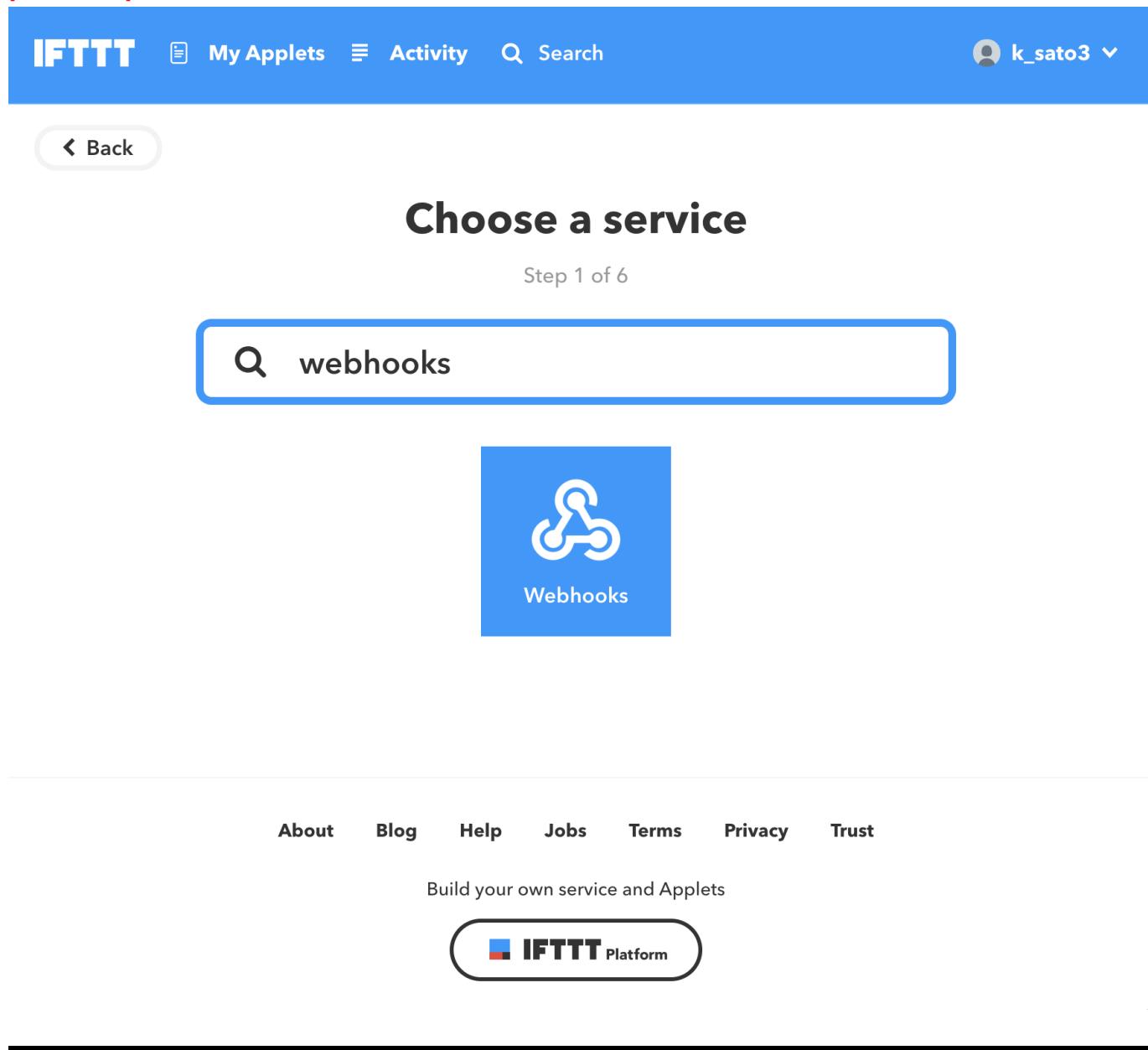
Build your own service and Applets

 **IFTTT** Platform

---

## (1) IFTTTにレシピを作成

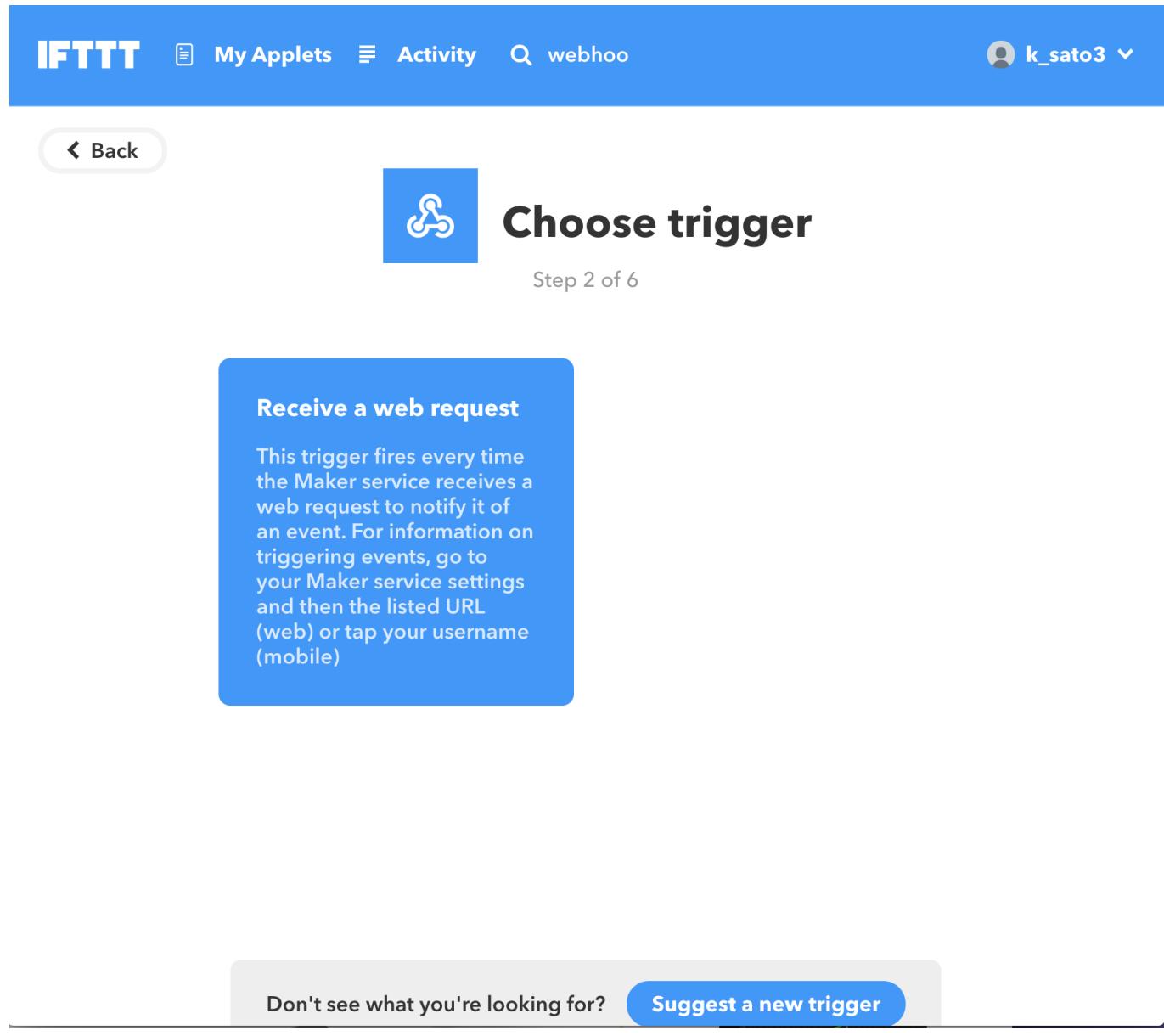
[Thisの設定] webhooksを検索し、webhooksのアイコンをクリック



The screenshot shows the IFTTT platform's "Choose a service" step. At the top, there are navigation links: "IFTTT", "My Applets", "Activity", "Search", and a user profile icon "k\_sato3". Below the search bar, it says "Step 1 of 6". A search bar contains the query "webhooks". A large blue button labeled "Webhooks" with a three-chain icon is prominently displayed. At the bottom, there are links for "About", "Blog", "Help", "Jobs", "Terms", "Privacy", and "Trust". A "Build your own service and Applets" link is also present, along with the "IFTTT Platform" logo.

(1) IFTTTにレシピを作成

[Thisの設定] 「Receive a web request」 をクリック



The screenshot shows the IFTTT interface for creating a new recipe. At the top, there's a blue header bar with the IFTTT logo, a search bar containing 'webhook', and a user profile icon labeled 'k\_sato3'. Below the header, a large button with a blue background and white text says 'Choose trigger'. Underneath this, a sub-section titled 'Receive a web request' is displayed, which includes a detailed description of the trigger: 'This trigger fires every time the Maker service receives a web request to notify it of an event. For information on triggering events, go to your Maker service settings and then the listed URL (web) or tap your username (mobile)'. At the bottom of the screen, there are two buttons: 'Don't see what you're looking for?' and 'Suggest a new trigger'.

(1) IFTTTにレシピを作成

[Thisの設定] Event Nameに Nefry\_IFTTT と記入し、「Create trigger」をクリック

The screenshot shows the IFTTT web interface. At the top, there are navigation links: 'IFTTT', 'My Applets', 'Activity', and a search bar 'webhoo'. On the right, a user profile icon is visible with the name 'k\_sato3'. Below the header, a back button '< Back' is on the left. In the center, there's a blue square icon with a white three-dot symbol. To its right, the title 'Complete trigger fields' is displayed in large bold letters, with 'Step 2 of 6' underneath it. A callout box contains the following text:

**Receive a web request**

This trigger fires every time the Maker service receives a web request to notify it of an event. For information on triggering events, go to your Maker service settings and then the listed URL (web) or tap your username (mobile)

**Event Name**

Nefry\_IFTTT

The name of the event, like "button\_pressed" or "front\_door\_opened"

**Create trigger**

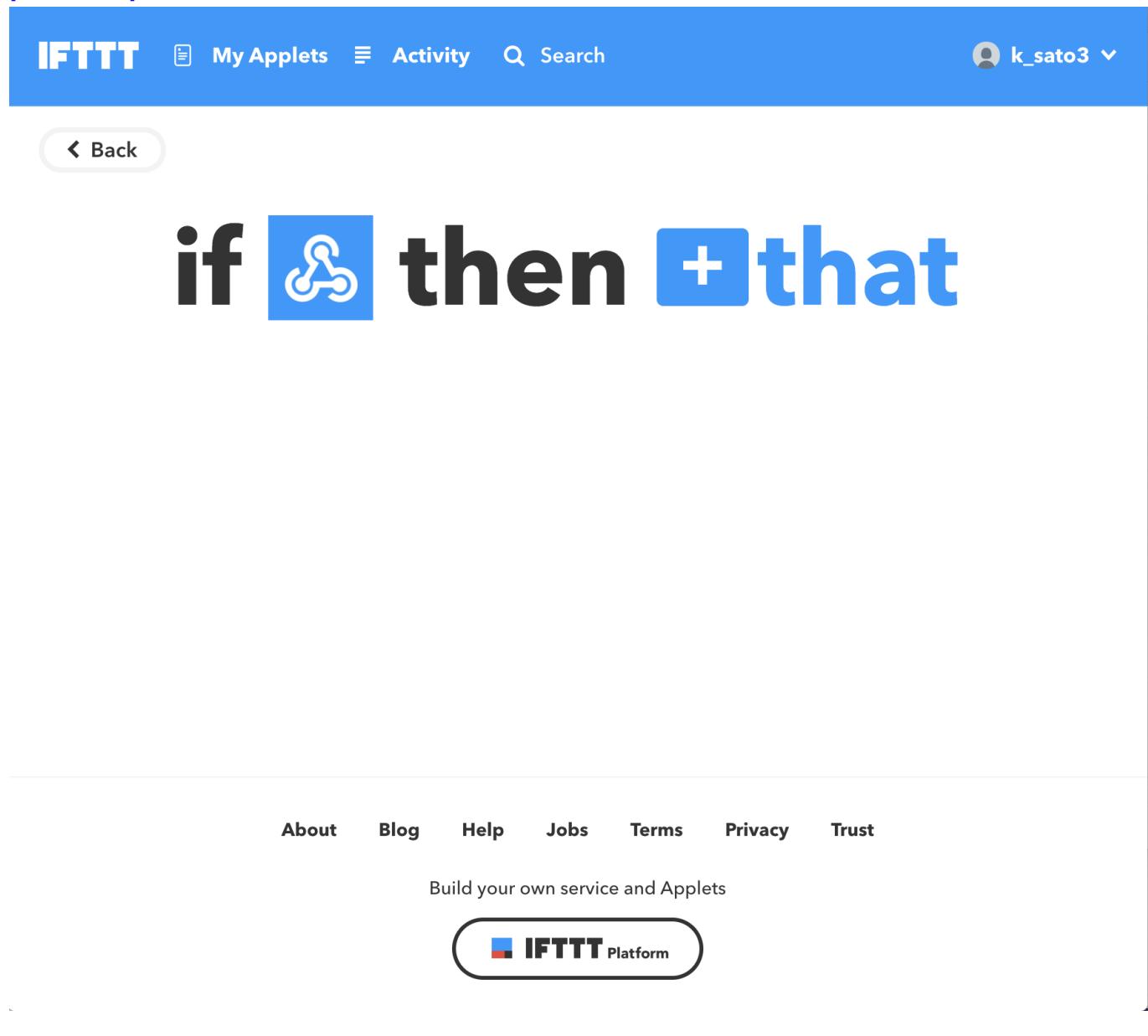
---

[About](#)   [Blog](#)   [Help](#)   [Jobs](#)   [Terms](#)   [Privacy](#)   [Trust](#)

---

(1) IFTTTにレシピを作成

[Thatの設定] 「+that」 をクリック



The screenshot shows the IFTTT platform's recipe creation interface. At the top, there's a blue header bar with the IFTTT logo, navigation links for 'My Applets', 'Activity', and 'Search', and a user profile icon for 'k\_sato3'. Below the header, a large central area displays the text 'if [app icon] then +that' in a bold, sans-serif font. To the left of 'if' is a back arrow labeled 'Back'. At the bottom of the main area, there are several small links: 'About', 'Blog', 'Help', 'Jobs', 'Terms', 'Privacy', and 'Trust'. Below these links is a button that says 'Build your own service and Applets'. At the very bottom is a prominent oval-shaped button with the text 'IFTTT Platform' and the IFTTT logo.

(1) IFTTTにレシピを作成

[Thatの設定] twitter を検索し、twitterのアイコンをクリック

The screenshot shows the IFTTT web interface. At the top, there is a blue header bar with the IFTTT logo, a 'My Applets' button, an 'Activity' button, a search bar containing 'webhoo', and a user profile icon labeled 'k\_sato3'. Below the header, there is a back button labeled '< Back'. The main title 'Choose action service' is displayed in bold black font, followed by 'Step 3 of 6'. A search bar contains the text 'twitter'. Below the search bar, a large blue square button features the white Twitter bird icon and the word 'Twitter'.

< Back

## Choose action service

Step 3 of 6

Q twitter



[About](#)   [Blog](#)   [Help](#)   [Jobs](#)   [Terms](#)   [Privacy](#)   [Trust](#)

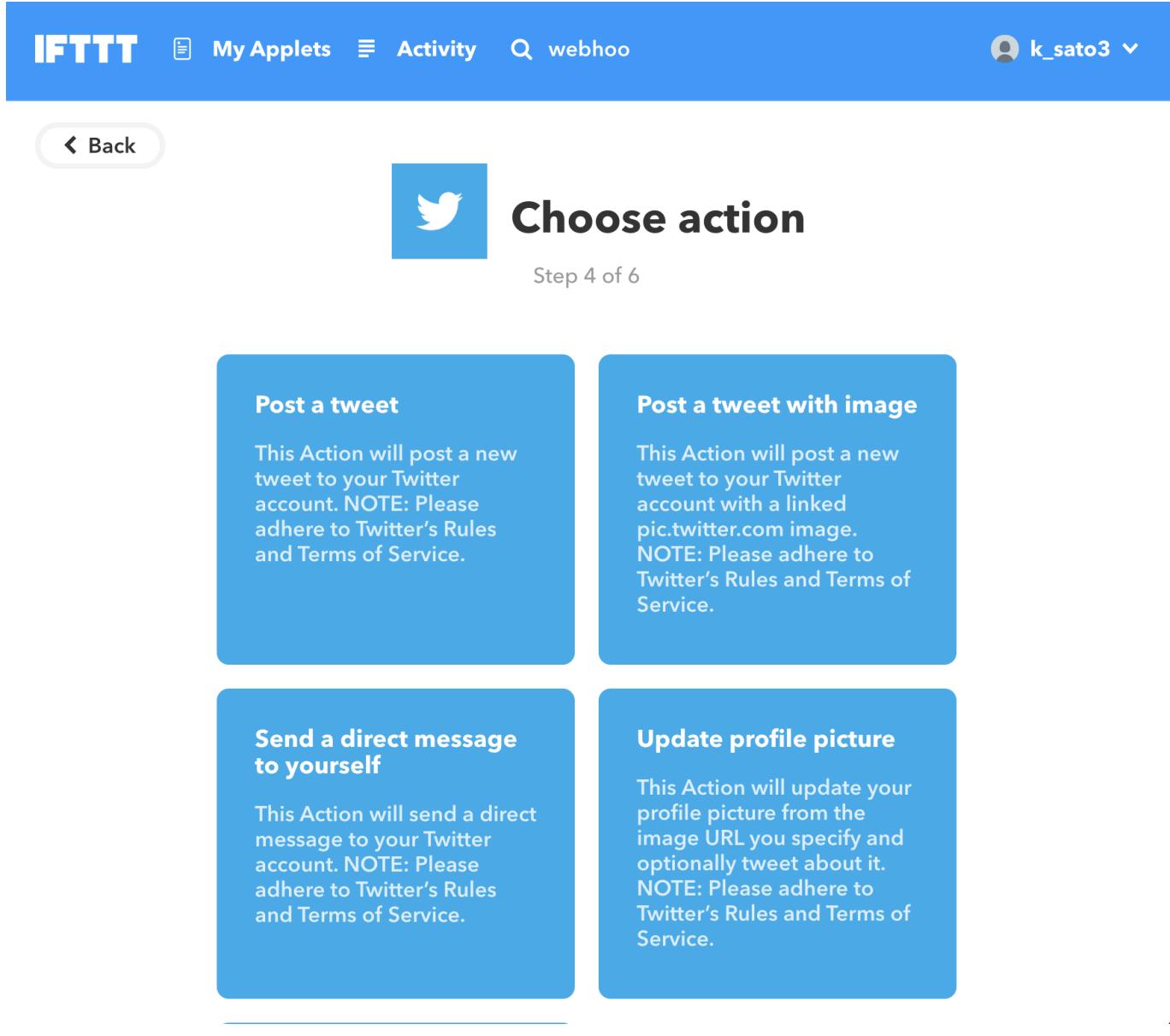
Build your own service and Applets

 **IFTTT** Platform

---

(1) IFTTTにレシピを作成

[Thatの設定] 「Post a tweet」をクリック



The screenshot shows the IFTTT interface for creating a new action. At the top, there's a blue header bar with the IFTTT logo, 'My Applets', 'Activity', a search bar with 'webhoo', and a user profile icon 'k\_sato3'. Below the header, a back button labeled 'Back' is visible. The main title 'Choose action' is centered above a progress indicator 'Step 4 of 6'. There are four blue rounded rectangular boxes, each representing a different action type:

- Post a tweet**: This Action will post a new tweet to your Twitter account. NOTE: Please adhere to Twitter's Rules and Terms of Service.
- Post a tweet with image**: This Action will post a new tweet to your Twitter account with a linked pic.twitter.com image. NOTE: Please adhere to Twitter's Rules and Terms of Service.
- Send a direct message to yourself**: This Action will send a direct message to your Twitter account. NOTE: Please adhere to Twitter's Rules and Terms of Service.
- Update profile picture**: This Action will update your profile picture from the image URL you specify and optionally tweet about it. NOTE: Please adhere to Twitter's Rules and Terms of Service.

## (1) IFTTTにレシピを作成

[Thatの設定]

Tweet text 欄にツイートしたい内容を入力し、(例: 【Bot】Nefry BTとIFTTT連携test) 「Create action」をクリック

The screenshot shows the IFTTT web interface. At the top, there's a blue header bar with the IFTTT logo, a 'My Applets' button, an 'Activity' section, a search bar containing 'webhoo', and a user profile icon labeled 'k\_sato3'. Below the header, there's a back button labeled 'Back'.

The main content area has a blue background. At the top left is a Twitter icon. To its right, the text 'Complete action fields' is displayed in large bold letters, with 'Step 5 of 6' underneath it. The central part of the screen is a white card with a rounded rectangle. Inside this card, the title 'Post a tweet' is at the top, followed by a note about posting to Twitter. Below that is a 'Tweet text' input field containing the text '[Bot] Nefry BTとIFTTT連携test!'. There are two buttons at the bottom of the card: 'Add ingredient' on the right and 'Create action' on the far right.

[About](#)   [Blog](#)   [Help](#)   [Jobs](#)   [Terms](#)   [Privacy](#)   [Trust](#)

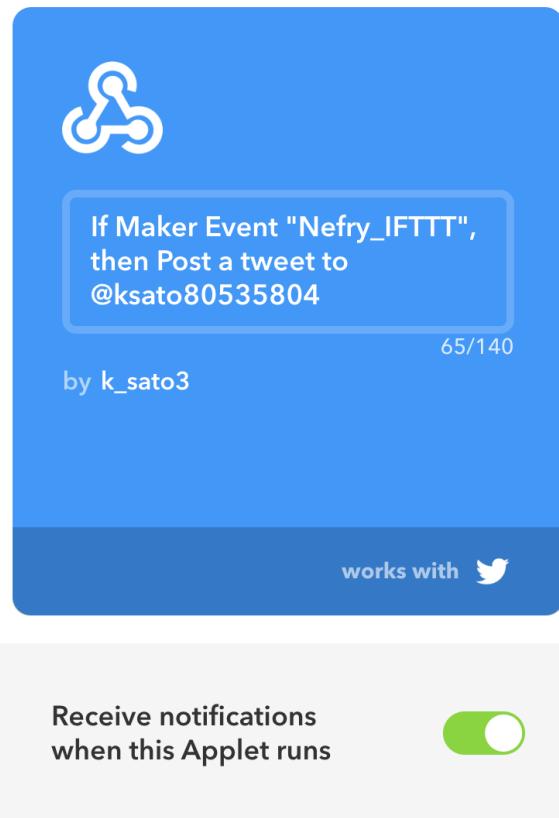
---

(1) IFTTTにレシピを作成

[Thatの設定] 「Finish」をクリック

## Review and finish

Step 6 of 6



これで、「**Nefryのボタンを押したらIFTTTのWebhooksにアクセスする**」レシピの作成完了です。

---

## (2) webhooksのSecret Key を取得

---

### (2) Secret Key を取得

webhooks で検索



My Applets

Activity

webhooks



k\_sato3

My Applets &gt; Webhooks

The image shows an IFTTT applet card. At the top is a blue header with the IFTTT logo. Below it is a white card area containing the applet's title and description. The title is "If Maker Event". The description is "Nefry\_IFTTT", "then Post a tweet to @ksato80535804". Below the title, it says "by k\_sato3". A large "On" button with a green circle is in the center. At the bottom, there are two sections: a list of stats ("Created on Jul 30 2018", "Never run") and a status message ("This Applet usually runs within an hour"). A "Check now" button is also present.

If Maker Event  
"Nefry\_IFTTT",  
then Post a tweet  
to  
@ksato80535804

by k\_sato3

On

- Created on Jul 30 2018
- Never run

This Applet usually runs within an hour

Check now

## (2) Secret Key を取得

「Services」をクリックし、webhooksのアイコンをクリック

The screenshot shows the IFTTT website interface. At the top, there is a navigation bar with the IFTTT logo, a 'My Applets' button, an 'Activity' button, a search bar containing the text 'Search', and a user profile icon labeled 'k\_sato3'. Below the navigation bar is a search bar with the text 'webhooks' typed into it. A blue rectangular border highlights the search bar and the results area. Underneath the search bar, there are two tabs: 'Applets' (which is currently selected) and 'Services'. In the main content area, there is a single card for 'Webhooks'. The card has a blue background with a white icon of three interconnected circles at the top, followed by the word 'Webhooks' below it.

## (2) Secret Key を取得

右上の「Settings」をクリック

The screenshot shows the IFTTT Webhooks page. At the top, there are navigation links for "My Applets", "Activity", "Search", and user profile "k\_sato3". Below the header is a large white "Webhooks" logo. A descriptive text block states: "Integrate other services on IFTTT with your DIY projects. You can create Applets that work with any device or app that can make or receive a web request. If you'd like to build your own service and Applets, check out the IFTTT platform." The main section is titled "My Applets" and displays a single active applet card. The card has a blue header with the IFTTT logo and the title "If Maker Event 'Nefry\_IFTTT', then Post a tweet to @ksato80535804". The body of the card contains the applet's logic: "If Maker Event 'Nefry\_IFTTT', then Post a tweet to @ksato80535804". Below the logic, it shows a green "On" status indicator and a "works with" badge for Twitter. At the bottom of the card is a "Get more" button.

## (2) Secret Key を取得

URLの末尾の自身のSecret Keyをコピー

IFTTT  My Applets  Activity  Search  k\_sato3 ▾

---

My Applets > Webhooks



**Webhooks settings**

[View activity log](#)

---

**Account Info**

Connected as: [REDACTED]

URL: <https://maker.ifttt.com/use/> [REDACTED]

Status: **active**

[Edit connection](#)

---

[Disconnect Webhooks](#)

---

About    Blog    Help    Jobs    Terms    Privacy    Trust

Build your own service and Applets



次に、コピーしたWebhooksの Secret Key を Nefry に登録します。

---

### (3)Nefry BTの設定ページでSecret Keyを登録

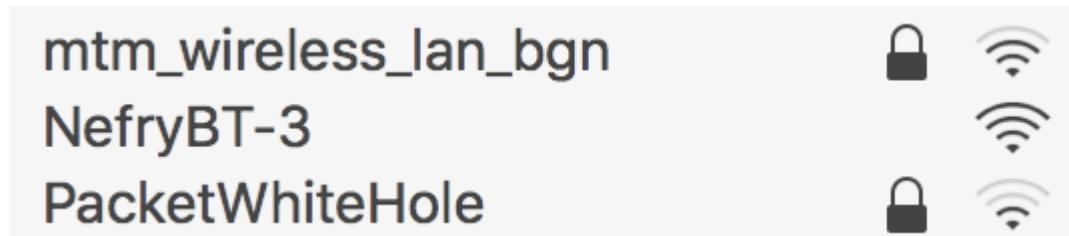
---

(3) Nefry BT の設定ページでSecret Key を登録

wifi設定

- Nefry BTを自分のPCに接続。

- "NefryBT-(数字)"というWi-Fiに接続.



Wi-Fiに接続すると自動で設定ページのウィンドウが立ち上がる。 (立ち上がらない場合はブラウザで http://192.168.4.1 にアクセス) このページから Nefry BT に関する様々な設定が可能。

Wifi Sport:  
IP Address: 192.168.4.1  
Module ID: NefryBT-FD3F

- [Setup WiFi](#)
- [Data Store](#)
- [Module Config](#)
- [Web Sketch Download](#)
- [Upload Sketch](#)

Nefry library:0.9.2  
Running ProgramName:

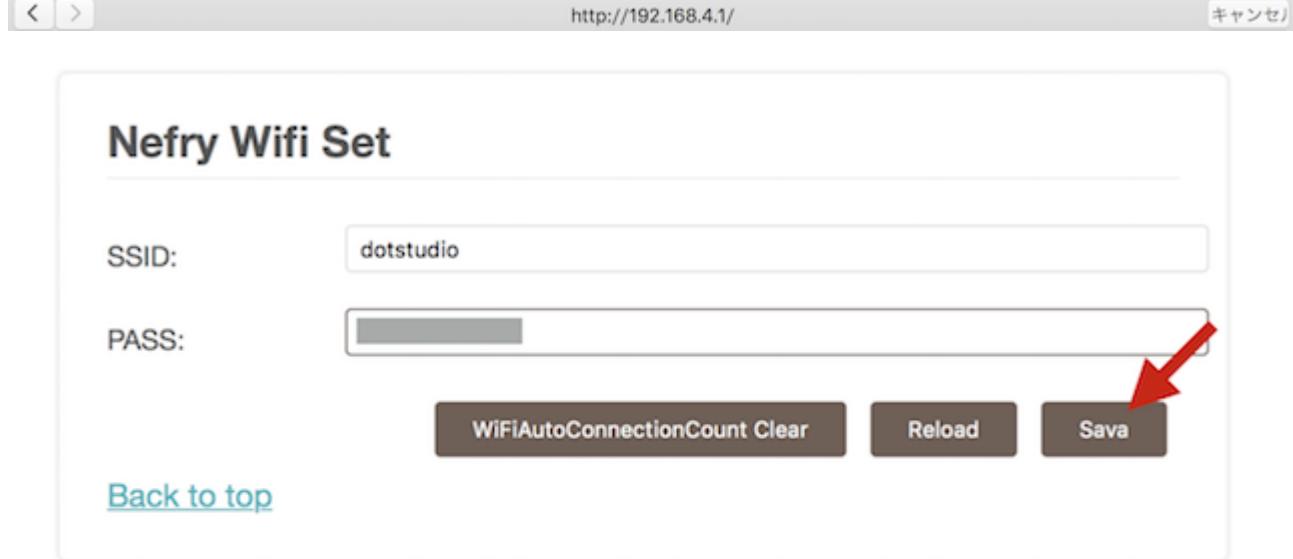
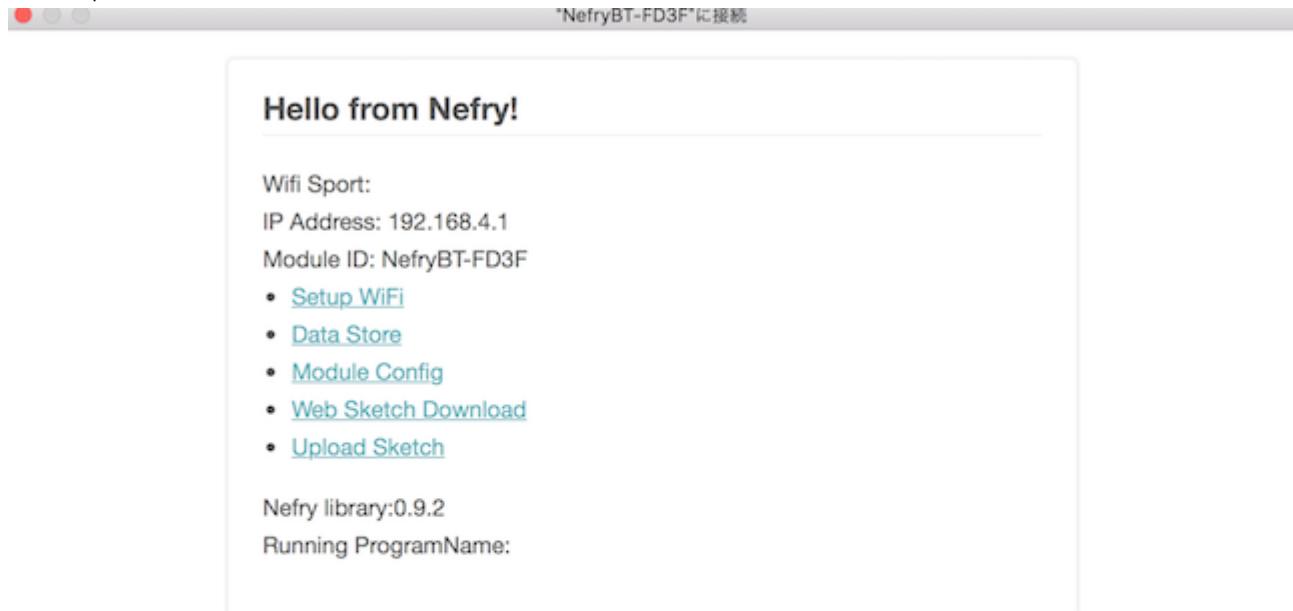


---

### (3) Nefry BT の設定ページでSecret Key を登録

wifi設定

「Setup WiFi」をクリックし、利用するWi-FiのSSIDとパスワードを入力し、「Save」で保存。



1. wx02-6dbc30 (-40)
2. dotstudio (-49)
3. ESP\_DEAA2D (-51)
4. 4CE676AD66EA (-72)
5. HUMAX-AF0D0 (-76)

再度、Nefry BTのWi-Fiにつなぎ、トップ画面を開くと、利用するWi-FiのSSIDとNefry BTのIPアドレスが表示される。

### (3) Nefry BT の設定ページでSecret Key を登録

トップ画面から「Data Store」に移動。

## Hello from Nefry!

Wifi Sport: dotstudio

IP Address: 192.168.0.133

Module ID: Nefry-FD3F

- [Setup WiFi](#)
- [Data Store](#)
- [Module Config](#)
- [Web Sketch Download](#)
- [Upload Sketch](#)

Nefry library:0.9.2

Running ProgramName:



### (3) Nefry BT の設定ページでSecret Key を登録

先ほど取得したWebhooksの「Secret Key」と「Event Name」をそれぞれ入力し、保存。

## Nefry DataStore Setup

このページはプログラム内から読み書きした値を表示、編集することができます。

わざわざプログラムを書き換えずに値を変更できるためWebサービスでアクセスキーが必要になる場合やモードを切り替える時など環境変数として扱うことができます。

- setStoreValue or setStoreStr : 値の保存
- getStoreValue or getStoreStr : 値の取得
- setStoreTitle : 内容の表示

それぞれの関数の使い方はNefry公式サイトをご覧になるか、サンプルプログラムを参考にしてください。

SecretKey

Event

NefryIFTTT

LINE Auth

LINEMessage

Save

[Back to top](#)

---

(3) Nefry BT の設定ページでSecret Key を登録

Nefry BTのボタンを押して試してみましょう！



Nefryのボタンを押すとTweetすることができました！（IFTTTのサーバ状況により送信に時間がかかる場合があります。）

## 2.LEDを光らせるプログラムを書いてみよう

<https://dotstud.io/docs/nefrybt-led/>

## 2.LEDを光らせるプログラムを書いてみよう

### (0)準備

- Arduino インストール <https://dotstud.io/docs/arduinoide-setup/>
- Arduino セットアップ <https://dotstud.io/docs/nefrybt-arduino-ide-setup/>

## 2.LEDを光らせるプログラムを書いてみよう

### (0)準備 Arduino インストール

Arduinoのサイトから自身の環境に合わせたダウンロードリンクを選択。次のページでドネーション（寄付金）を求められますが、特に気にせず「JUST DOWNLOAD」を選択し、インストール。



## Download the Arduino IDE

**ARDUINO 1.8.3**  
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.  
This software can be used with any Arduino board.  
Refer to the [Getting Started](#) page for Installation instructions.

**Windows** Installer  
**Windows** ZIP file for non admin install  
**Windows app** Get

**Mac OS X 10.7 Lion or newer** (highlighted)

**Linux** 32 bits  
**Linux** 64 bits  
**Linux** ARM

[Release Notes](#)  
[Source Code](#)  
[Checksums \(sha512\)](#)

ARDUINO SOFTWARE  
HOURLY BUILDS

ARDUINO 1.0.6 / 1.5.x / 1.6.x  
PREVIOUS RELEASES

## Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)

SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED  
17,157,203 TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND  
GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE  
USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES,  
CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT  
WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!

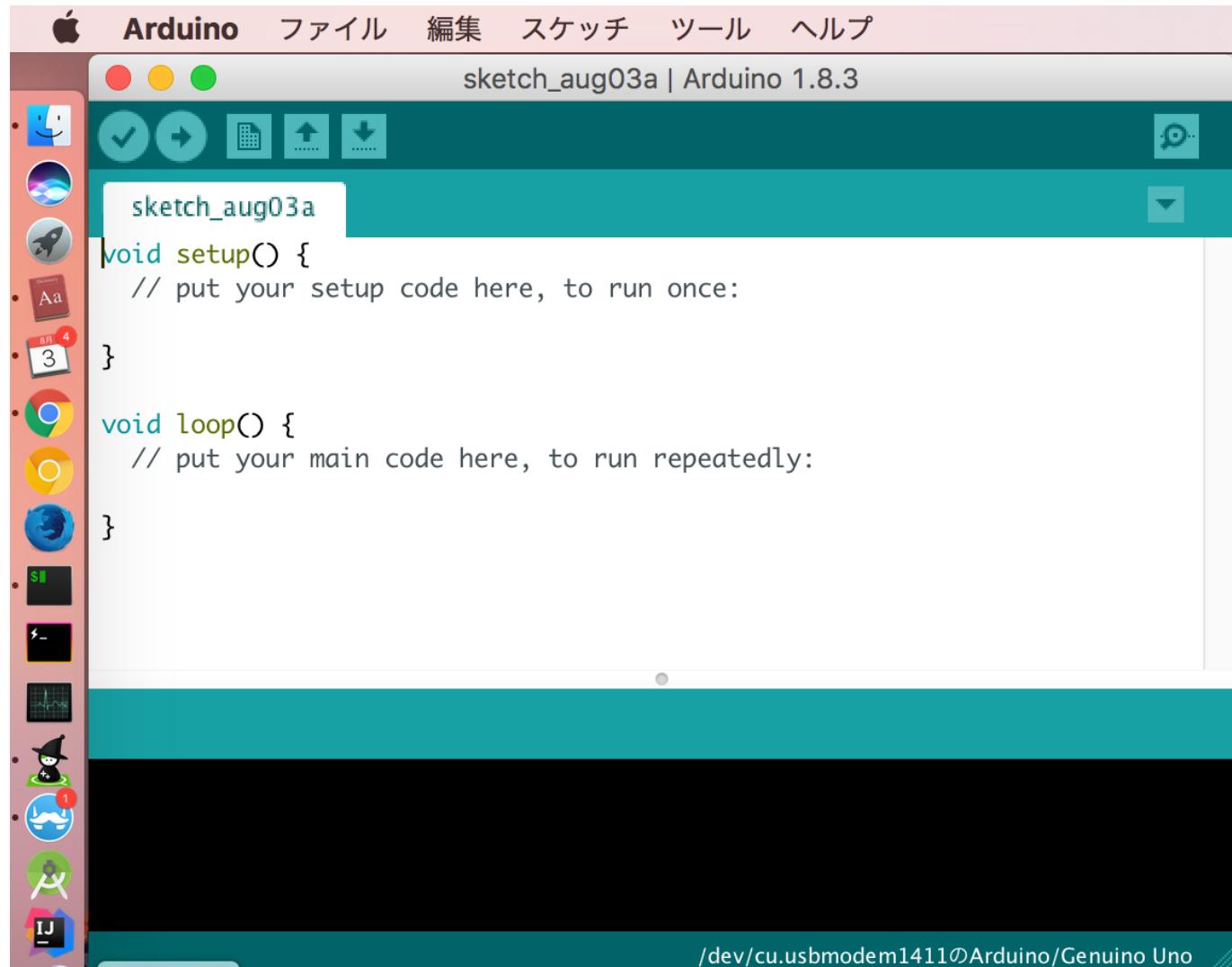
\$3    \$5    \$10    \$25    \$50    OTHER

JUST DOWNLOAD (highlighted) CONTRIBUTE & DOWNLOAD

## 2.LEDを光らせるプログラムを書いてみよう

(0)準備 Arduino インストール

Arduinoファイルを起動し、下記のような画面が表示されれば起動完了。

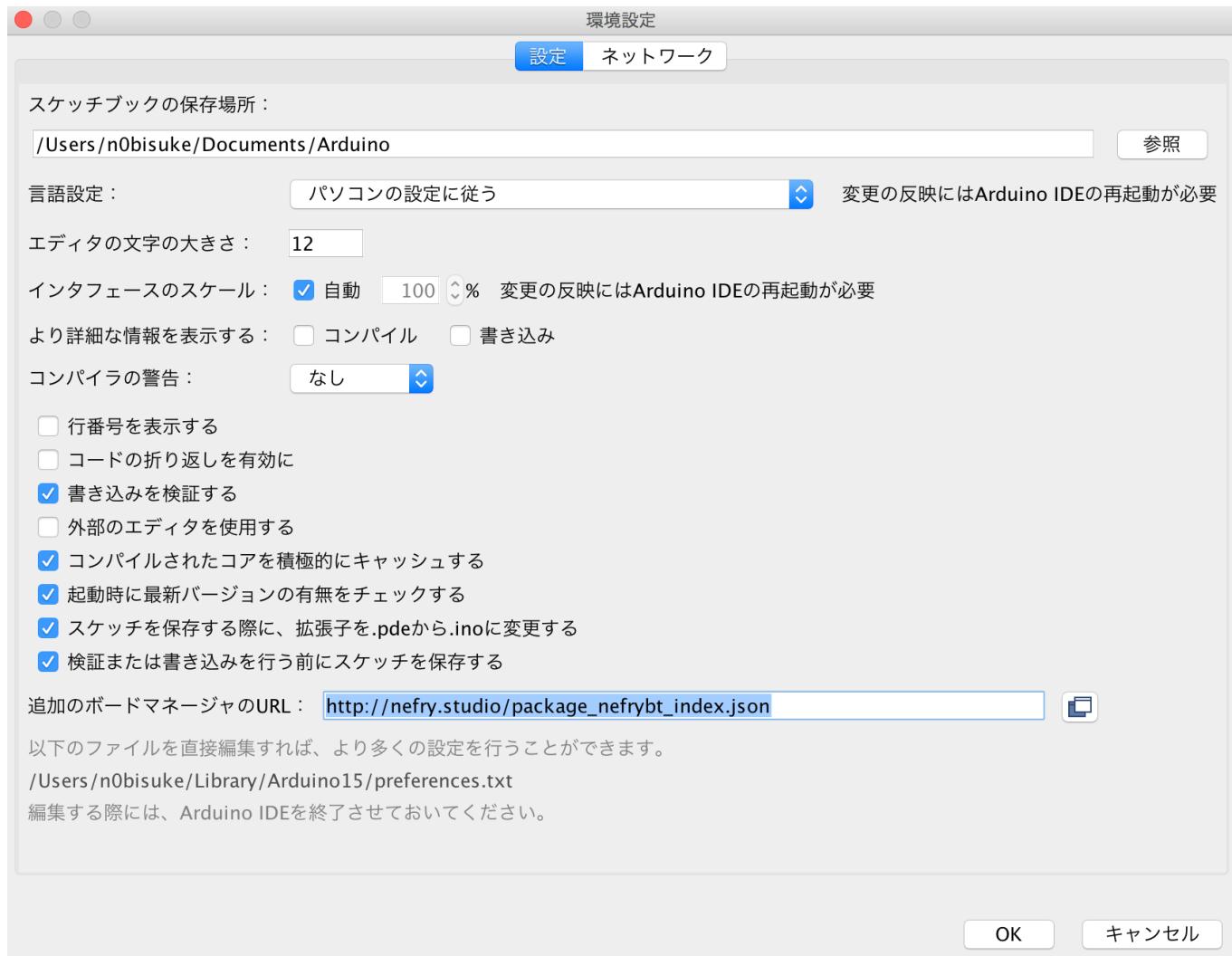


## 2.LEDを光らせるプログラムを書いてみよう

### (0)準備 Arduino セットアップ

メニューのArduino > Preferencesを選択。追加のボードマネージャのURLのフォームに以下のURLを追加し、OKを選択。

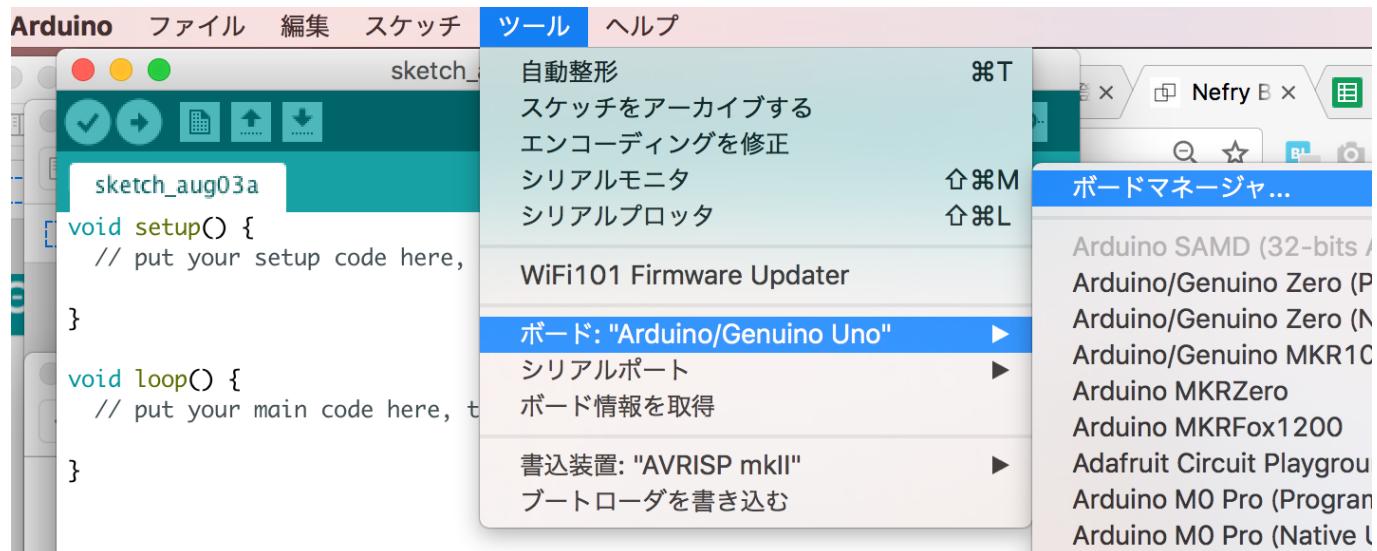
[http://nefry.studio/package\\_nefrybt\\_index.json](http://nefry.studio/package_nefrybt_index.json)



## 2.LEDを光らせるプログラムを書いてみよう

### (0)準備 Arduino セットアップ

メニューのツール > ボード > ボードマネージャを選択。



## 2.LEDを光らせるプログラムを書いてみよう

## (0)準備 Arduino セットアップ

フォームにNefryと入力して検索する。



バージョン1.4.0を選択し、インストールを押す(バージョンは1.2.2以上のものを選択)

## 2.LEDを光らせるプログラムを書いてみよう

### (1)ボードの選択

メニューのツール > ボード > Nefry BT R2/R3を選択。

The screenshot shows the Arduino IDE interface. On the left, there's a sidebar with links like 'index.md', 'Nefry BTとIFT', '保護された通信', 'BLOG', and 'DOCS'. The main area has a heading 'Nefry BTで内蔵LEDでLチ' and some text about controlling an LED via Bluetooth. On the right, the 'Tools' menu is open, showing various options like '自動整形', 'WiFi101 Firmware Updater', and a list of boards. The 'Board' section is expanded, showing 'Nefry BT R2 / R3' as the selected option. Below the board list, the 'Nefry(ESP32)Module' is listed as installed.

## 2.LEDを光らせるプログラムを書いてみよう

### (2)Nefry BTをPCのUSBポートに挿入

- Nefry BTをPCのUSBポートに挿入.
- メニューのツール >シリアルポート > /dev/cu.usbserial-xxxxxx を選択. (環境によっては /dev/tty.usbserial-xxxxxxなどの場合あり) ※Windowsだと表記が異なる可能性あり

## 2.LEDを光らせるプログラムを書いてみよう

### (3)プログラムの書き込み

Arduinoを選択し、メニューのファイル > 新規ファイルを選択。



初期状態で書かれているコードを削除し、以下のコードに差し替える

```

#include <Nefry.h>
//フルカラーLED ランダムにカラーが変わります.
#define SEED_PIN A0

void setup() {
  randomSeed(analogRead(SEED_PIN));
}

int red,green,blue;
void loop() {
  red=random(255); //random関数は0-255の数値をランダムに返します.
  green=random(255);
  blue=random(255);
  Nefry.setLed(red,green,blue); //LEDがランダムに点灯します.
  String color="Red:";color+=red;
}

```

```
color+=" Green:";color+=green;
color+=" Blue:";color+=blue;
Nefry.ndelay(1000); //1秒待つ
}
```

※A0を指定していますが、乱数のSeed用に使っているだけで、内蔵のLEDがA0という訳ではありません。

## 2.LEDを光らせるプログラムを書いてみよう

### (4) プログラムの保存とボードに書き込み

左上の→ボタンを押し、ボード（Nefry BT）にプログラムを書き込む。



## 2.LEDを光らせるプログラムを書いてみよう

### (5) 確認

無事にプログラム書き込みが終わると、Nefry BTに内蔵してあるLEDがカラフルに光ります。

※プログラムが書き込めない場合は、設定を以下のように変えてみましょう

ツール > Upload Speed : 460800

## 3.光センサの情報を取得してfirebaseに記録してみよう

firebaseとは

BaaS (Backend as a Service) の1つ。Webアプリケーションやモバイルアプリケーションのバックエンドで行う機能を提供するクラウドサービス。実装者がバックエンド側の実装をすることなく、決められたAPIを叩くだけでクラウド上に用意された機能群を使うことが出来る。

使用するセンサ：光センサ



---

### 3.光センサの情報を取得してfirebaseに記録してみよう

(1) 光センサの値を確認する

Nefry BTのA2 と書かれたソケットに光センサを接続する。



---

Arduinoを起動し、メニューのファイル > 新規ファイルを選択。新規ファイルに以下のプログラムを書き込み、左上の→ボタンを押し、実行。

```
#include <Nefry.h>
#define PIN A2

void setup()
{
    pinMode(PIN, INPUT);
}

void loop()
{
    long sum = 0;
```

```
// 32回分の合計をする
for(int i=0; i<32; i++)
{
    sum += analogRead(PIN);
}
// 合計を割って平均を取る
sum >>= 5;
Nefry.println(sum);
Nefry.ndelay(10);
}
```

### 3.光センサの情報を取得してfirebaseに記録してみよう

#### (1) 光センサの値を確認する

- ツール > シリアルモニタ(command+shift+M) を選択し、 115200bpsに設定すると値を確認できる。
- ツール > シリアルプロット(command+shift+L) を選択し、 115200bpsに設定するとグラフで可視化される ※注意：同時に2つ開くことはできない ※音センサも同じプログラムで実行可能

#### (2) Firebaseへの記録

1. Firebase を開く <https://firebase.google.com/?hl=ja>

2. スタートガイドをクリック

The screenshot shows the Firebase homepage. At the top, there's a navigation bar with the Firebase logo, product links, usage examples, pricing, documentation, search, and a 'Move to console' button. Below the navigation, there's a banner for 'Firebase Crashlytics' with the tagline 'Prioritize and fix issues with powerful, realtime crash reporting.' A large graphic of a smartphone displaying a lock, a dollar sign, and a barcode is shown. The main headline reads 'Firebase helps mobile app teams succeed.' Below the headline are two buttons: 'スタートガイド' and '動画を見る'. At the bottom, there are three columns of text: 'アプリをすばやく作成、インフラストラクチャの管理は不要', 'Google のインフラが支える、多数の人気アプリが信頼するサービス', and '連係する全プロダクトを1つのコンソールで管理'. The footer of the page also contains some text.

## (2) Firebaseへの記録

3.
  - プロジェクト名 nefry
  - アナリティクスと課金の地域 日本 と設定し、プロジェクトを作成



## (2) Firebaseへの記録

4. 「Database」を選択し、「Realtime Database」を選択

The screenshot shows the Firebase Project Overview page for a project named 'nefry'. The main header says 'Firebase へようこそ ここから始めましょう。' (Welcome to Firebase, let's start from here). It features three circular icons: 'iOS' (with text 'iOS アプリに Firebase を追加'), 'Android' (with text 'Android アプリに Firebase を追加'), and 'Web' (with text 'ウェブアプリに Firebase を追加'). Below this, there's a banner for 'Firebase の機能' (Features) with three cards: 'Analytics' (illustration of a person looking at a chart), 'Authentication' (illustration of user profiles), and 'Database' (illustration of server racks). At the bottom, there's a section for 'Cloud Firestore' with a large image of servers and databases, and another for 'Realtime Database' with a similar server illustration.

**Cloud Firestore**

Cloud Firestore の料金  
Cloud Firestore で実現できること

**Realtime Database**

Firebase 独自のデータベース。Cloud Firestore と同様に、リアルタイムのデータ同期をサポートしています。

**Functions**  
Firebase 機能の拡張と接続

**Storage**  
ユーザー作成コンテンツの保存と取得

**Hosting**  
ウェブアプリを秒単位でデプロイ

## (2) Firebaseへの記録

5. 「テストモードで開始」を選択し、「有効にする」をクリック

## (2) Firebaseへの記録

6.databaseのURLをコピーしておく

## (2) Firebaseへの記録

7. Nefryに以下のコードを書き込む

FIREBASE\_HOSTの部分は先ほどコピーしたものを記述(https:// は除く)

```
1 <html>
2 <body>
3   <p>
4     <div id="sensor"></div>
5   </p>
6   <script src="https://www.gstatic.com/firebasejs/4.12.1.firebaseio.js"></script>
7   <script>
8     var config = {
9       apiKey: "nefry-cef61",
10      authDomain: "projectId.firebaseio.com",
11      databaseURL: "https://xxxx.firebaseio.com", // この値を変更する
12      storageBucket: "bucket.appspot.com"
13    };
14    firebase.initializeApp(config);
15    var database = firebase.database();
16    var dataRef = database.ref('Nefry');
17    dataRef.once("value")
18      .then(function (snapshot) {
19        document.getElementById("sensor").innerHTML = snapshot.child("sensor").val();
20      });
21    </script>
22  </body>
23 </html>
```

```
#include <Nefry.h>
#include <NefryFireBase.h>
#define PIN A2
#define FIREBASE_HOST "xxxx.firebaseio.com"

NefryFireBase firebase;

void setup()
{
  pinMode(PIN, INPUT);
  firebase.begin(FIREBASE_HOST);
}

void loop()
{
  DataElement elem = DataElement();
  elem.setValue("sensor", analogRead(PIN));
  firebase.write("Nefry", &elem); //FireBaseのデータを書き込む
  Nefry.ndelay(1000);
}
```

---

## (2) Firebaseへの記録

## 8. 光センサの値がFireBaseでリアルタイムに更新されるのが確認できる



The screenshot shows the Firebase Realtime Database console. The left sidebar has icons for Home, Settings, and other database-related options. The main area is titled 'Database' and 'Realtime Database'. At the top, there's a URL bar with 'https://nefry-b2994.firebaseio.com/' and a note: 'セキュリティ ルールが公開として定義されているため、誰でもデータベース内のデータを窃取、変更、削除できます。' (Security rules are publicly defined, so anyone can read, write, or delete data from the database). Below this, the database structure is shown: 'nefry-b2994' > 'Nefry' > 'sensor: 2295'. There are also 'Dokumentに移動' (Move to document), '+', '(-)', and '...' buttons at the top right of the preview area.

## (3) webページを作成し、ブラウザで光センサの値を表示

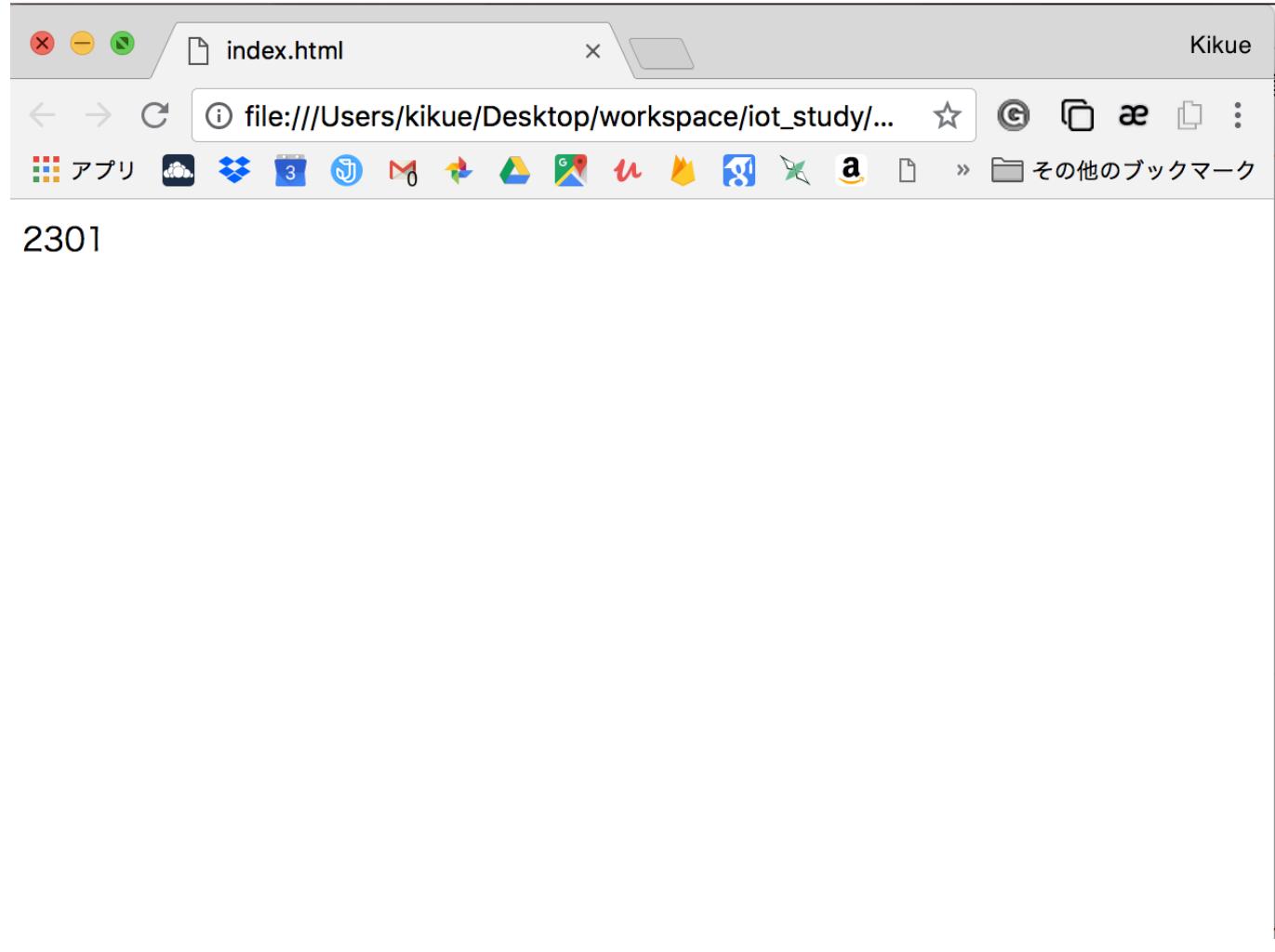
以下のソースコードのdatabaseURLを変更し、htmlページを作成。

```
<html>
<body>
    <p>
        <div id="sensor"></div>
    </p>
    <script src="https://www.gstatic.com/firebasejs/4.12.1/firebase.js">
    </script>
    <script>
        var config = {
            apiKey: "nefry-cef61",
            authDomain: "projectId.firebaseio.com",
            databaseURL: "https://xxxx.firebaseio.com", // この値を変更する
            storageBucket: "bucket.appspot.com"
        };
        firebase.initializeApp(config);
        var database = firebase.database();
        var dataRef = database.ref('/Nefry');
        dataRef.once("value")
            .then(function (snapshot) {
                document.getElementById("sensor").innerHTML =
                snapshot.child("sensor").val();
            });
    </script>
</body>
</html>
```

---

(3) webページを作成し、ブラウザで光センサの値を表示

Command+R or 左上の更新ボタンで更新すると、光センサの値が確認できる。



---

## 付録(他のセンサの使い方)

---

光センサ以外のセンサーヤや出力装置のサンプルコードを紹介 サンプルによってはArduino IDEに外部ライブラリのインストールが必要

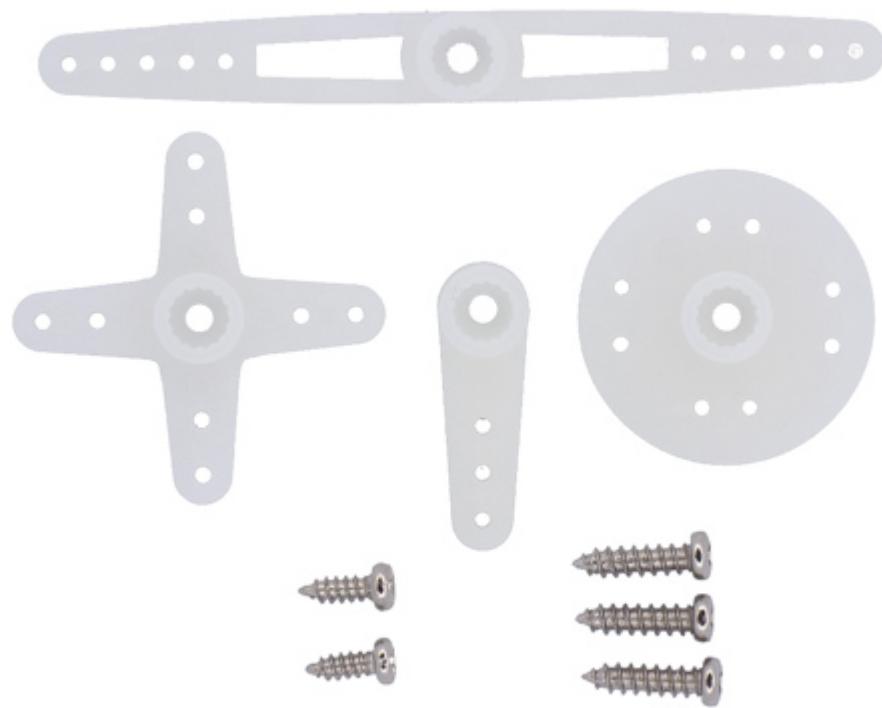
参考ページ <http://wiki.seeedstudio.com/Sensor/>

---

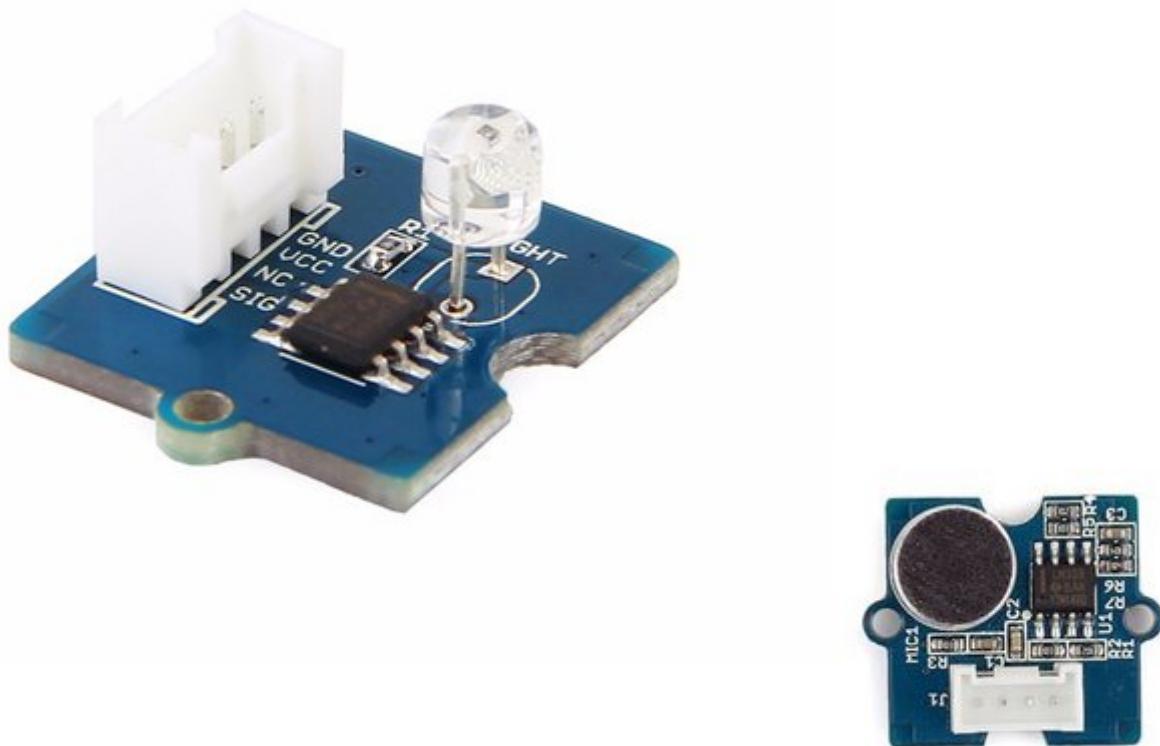
センサー一覧 <http://wiki.seeedstudio.com/>

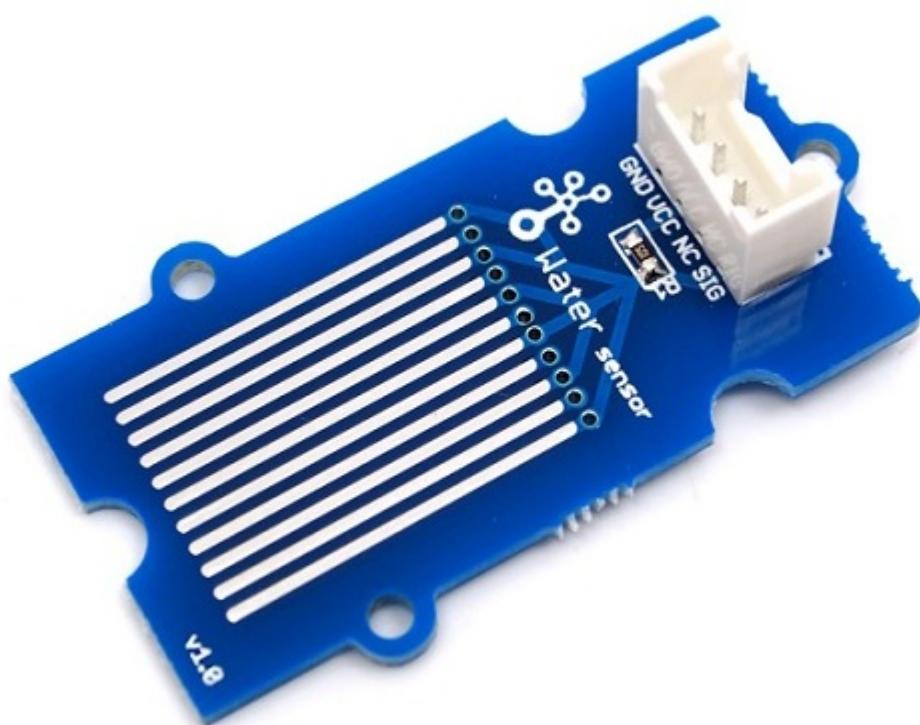
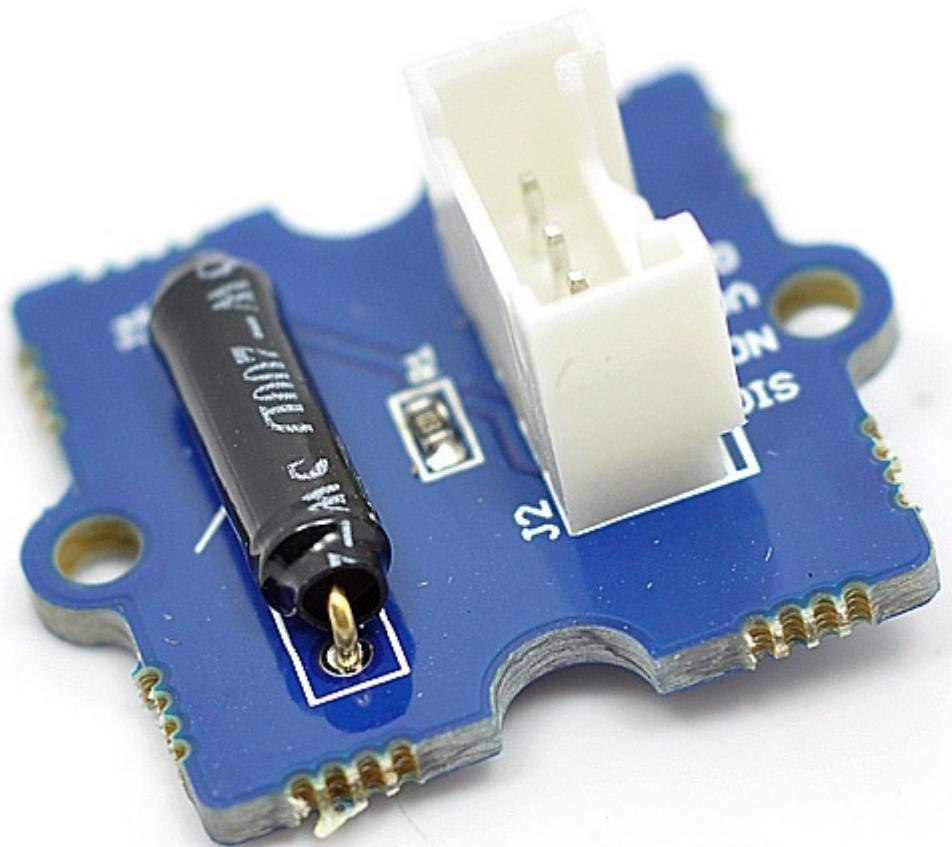


サー ボ モ タ



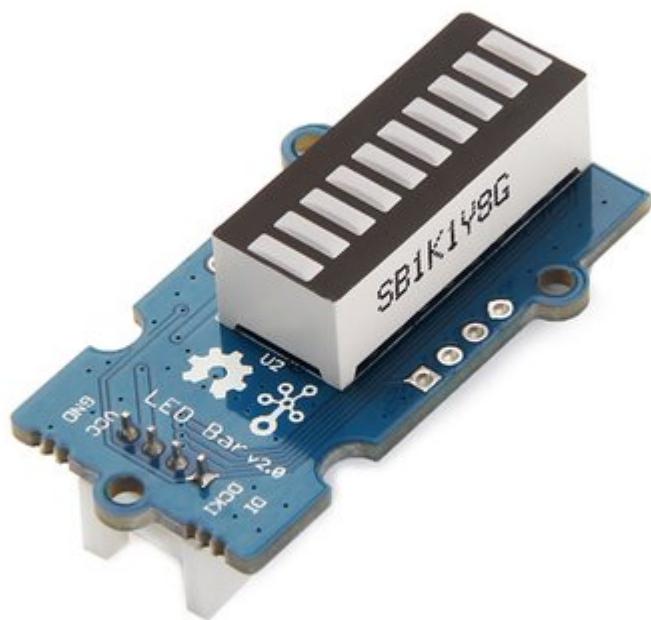
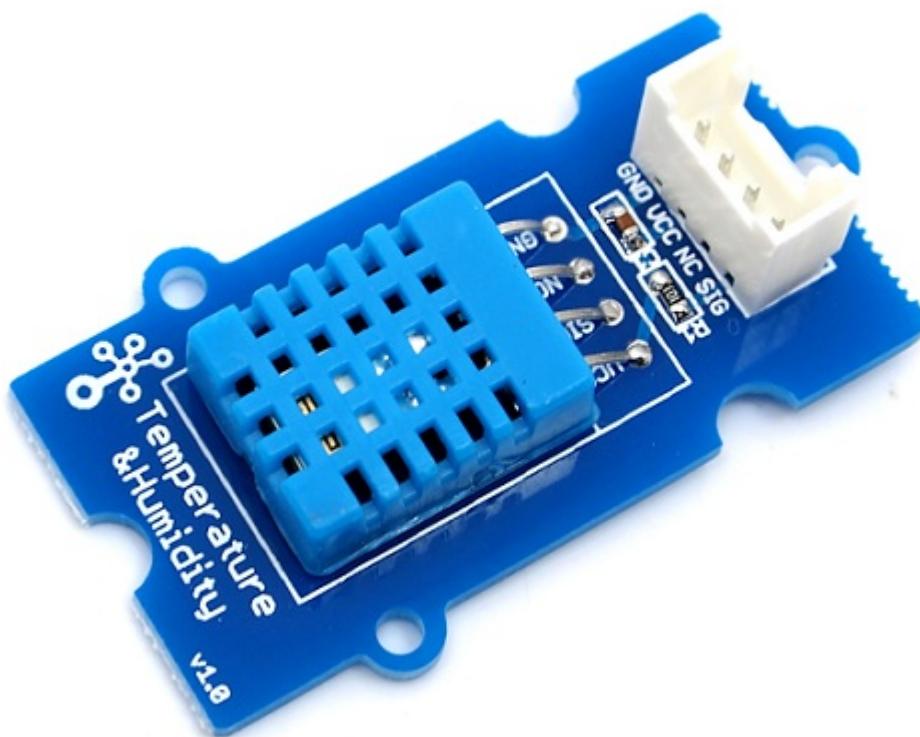
光センサ/音センサ/傾きセンサ/水センサ





モーションセンサ/温度湿度センサ/LEDbarセンサ

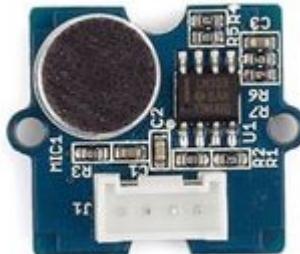




その他→ <https://dotstud.io/docs/grove/>

---

## 音センサ

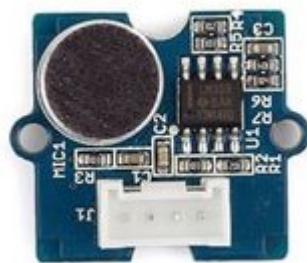
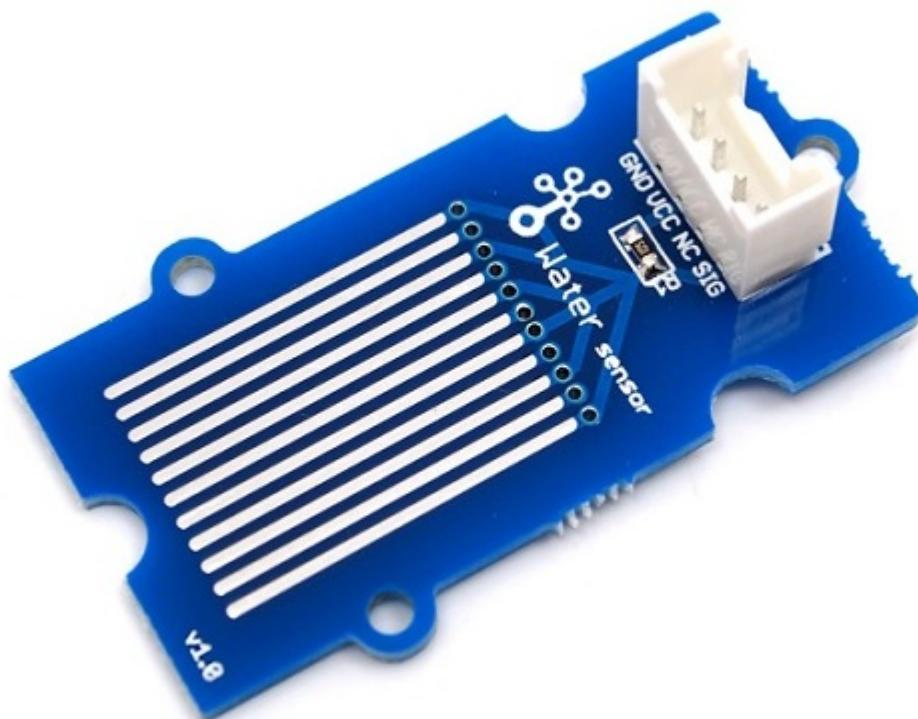
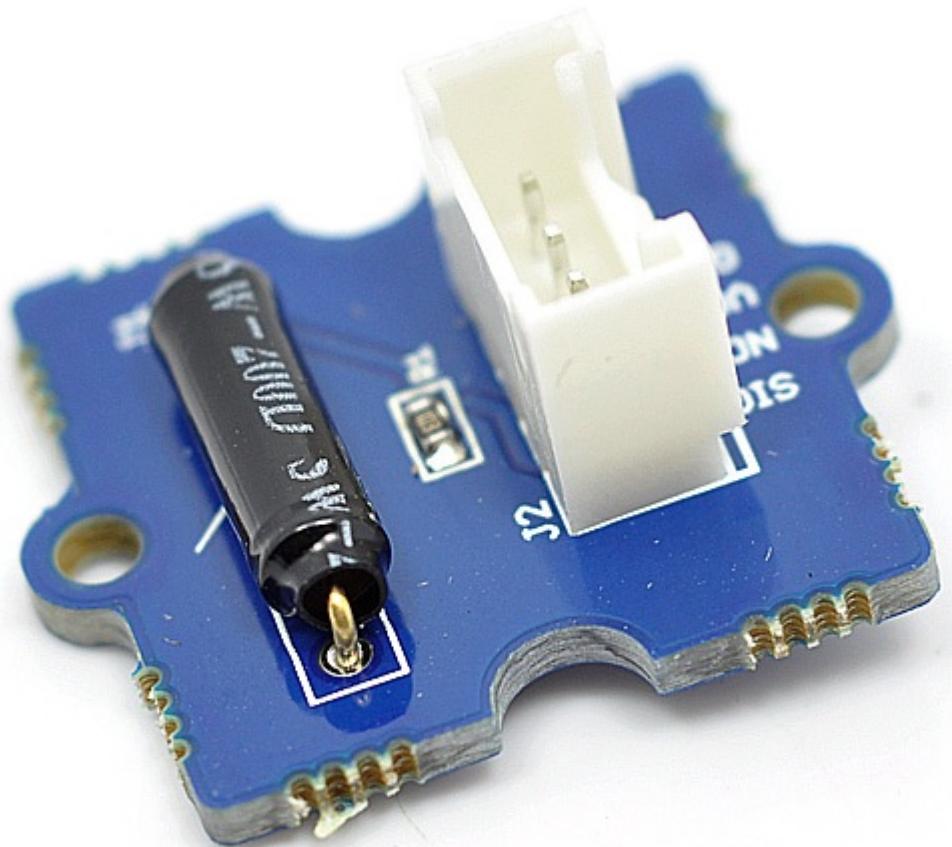


[http://wiki.seeedstudio.com/Grove-Sound\\_Sensor/](http://wiki.seeedstudio.com/Grove-Sound_Sensor/) A0に接続し、下記プログラムを実行。ツール>シリアルモニタ or シリアルプロットで値を確認できる

```
const int pinAdc = A0;
void setup()
{
    Serial.begin(115200);
}
void loop()
{
    long sum = 0;
    for(int i=0; i<32; i++)
    {
        sum += analogRead(pinAdc); //32回計測
    }
    sum >>= 5; // 平均を算出
    Serial.println(sum);
    delay(10); //10ms周期で計測
}
```

---

## 傾向センサ/水センサ(/音センサ)



```
#define PIN D2

void setup()
{
    Serial.begin(115200);
    pinMode(PIN, INPUT);
}

void loop()
{
    int v = digitalRead(PIN);
    Serial.println(v);
    delay(10);
}
```

ツール > シリアルモニタ or シリアルプロットで値を確認できる

---

### モーションセンサ (動きのある人を検知)



```
#define PIN D2

void setup()
```

```
{  
  Serial.begin(115200);  
  pinMode(PIN,INPUT);  
}  
  
void loop()  
{  
  if(digitalRead(PIN))//if it detects the moving people?  
    Serial.println("Hi,people is coming");  
  else  
    Serial.println("Watching");  
  
  delay(200);  
}
```

---

## サーボモータ

```

#define PIN D2
#include <Nefry.h> // Nefryのライブラリをインクルード
// LEDCのパラメータ設定
// LEDC_CHANNEL : チャンネル : 0
// LEDC_RESOLUTION_BITS: 目盛数 : 10bit (0~1023)
// LEDC_FREQUENCY : 周波数 : 50Hz (= 20ms周期)
#define LEDC_CHANNEL 0
#define LEDC_RESOLUTION_BITS 10
#define LEDC_FREQUENCY 50

// 30/1024*20ms = 0.59ms
// 77/1024*20ms = 1.50ms
// 122/1024*20ms = 2.38ms
uint32_t pulse[3] = {30, 77, 122};
int n = 0;

void setup() {
    ledcSetup(LEDC_CHANNEL, LEDC_FREQUENCY, LEDC_RESOLUTION_BITS);
    ledcAttachPin(PIN, LEDC_CHANNEL);
    Serial.println(pulse[n]);
    ledcWrite(0, pulse[n]);
    Nefry.enableSW();
}

void loop() {
    if (Nefry.readSW()) {
        n = (n+1) % 3;
        Serial.println(pulse[n]);
        ledcWrite(0, pulse[n]);
    }
}

```

```

#include <Nefry.h> // Nefryのライブラリをインクルード
#define PIN D2
// LEDCのパラメータ設定
#define LEDC_CHANNEL 0          // チャンネル : 0
#define LEDC_RESOLUTION_BITS 10 // 目盛数 : 10bit (0~1023)
#define LEDC_FREQUENCY 50       //周波数 : 50Hz (= 20ms周期)

// 30/1024*20ms = 0.59ms
// 77/1024*20ms = 1.50ms
// 122/1024*20ms = 2.38ms
uint32_t pulse[3] = {30, 77, 122};
int n = 0;

```

```

void setup() {
    ledcSetup(LEDC_CHANNEL, LEDC_FREQUENCY, LEDC_RESOLUTION_BITS);
    ledcAttachPin(PIN, LEDC_CHANNEL);
    Serial.println(pulse[n]);
    ledcWrite(0, pulse[n]);
    Nefry.enableSW();
}

void loop() {
    if (Nefry.readSW()) {
        n = (n+1) % 3;
        Serial.println(pulse[n]);
        ledcWrite(0, pulse[n]);
    }
}

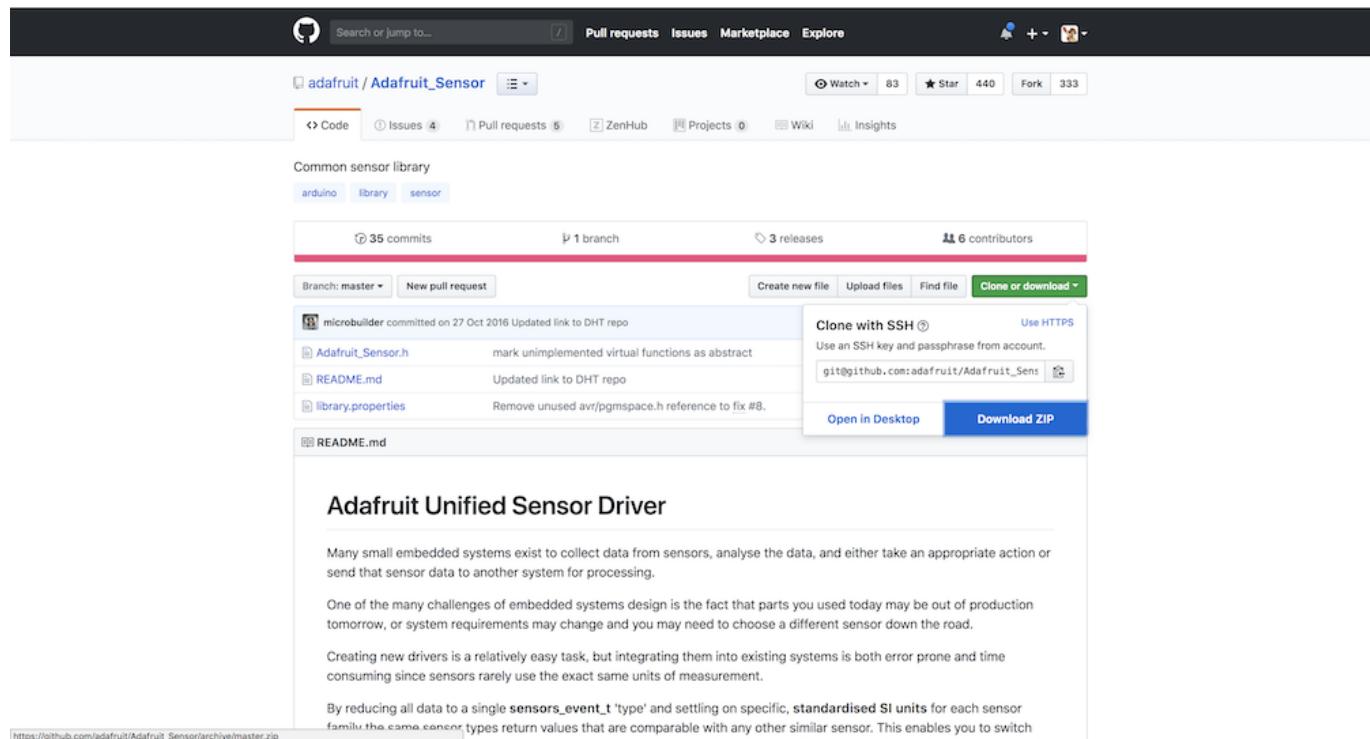
```

## 温度湿度センサ

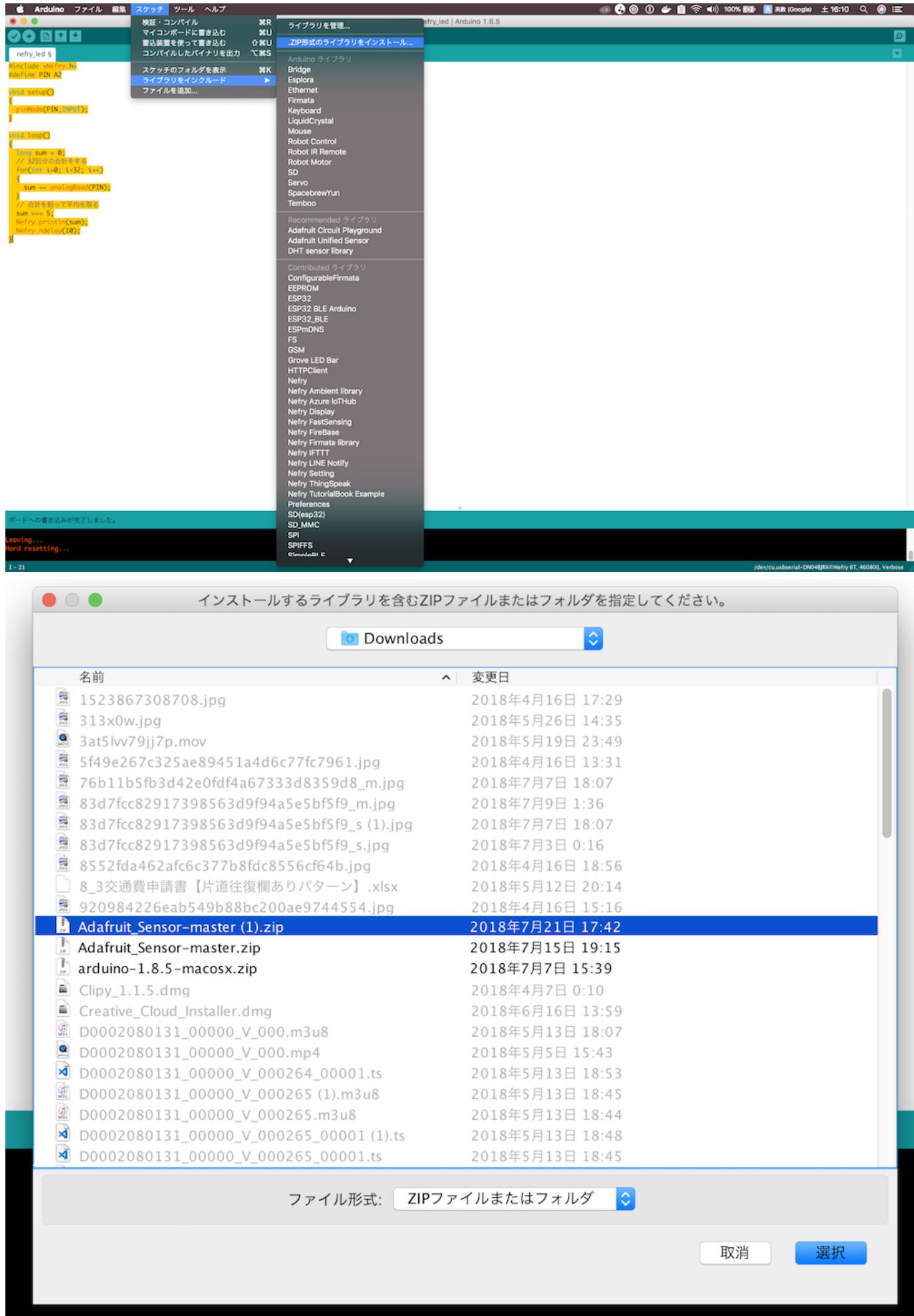
以下のライブラリをインストールする [https://github.com/adafruit/Adafruit\\_Sensor](https://github.com/adafruit/Adafruit_Sensor)  
<https://github.com/adafruit/DHT-sensor-library>

## 外部ライブラリのインポートの方法

1.zip形式でダウンロード



2.Arduino IDEを起動し、スケッチ > ライブラリをインクルード > ZIP形式のライブラリをインストール



1%

```
#include <NeFry.h> // NeFryのライブラリをインクルード

#include "DHT.h"
#define DHTPIN D2      // 値をD4に変更（接続するピンに応じて変更）

#define DHTTYPE DHT11 // DHT 11に変更

DHT dht(DHTPIN, DHTTYPE);

void setup() {
    //Serial.begin(9600); 削除
    NeFry.println("DHTxx test!"); // SerialをNeFryへ変更
    dht.begin();
}

void loop() {
    NeFry.ndelay(2000); // Serial.delayをNeFry.ndelayへ変更

    float h = dht.readHumidity();
    float t = dht.readTemperature();
    float f = dht.readTemperature(true);

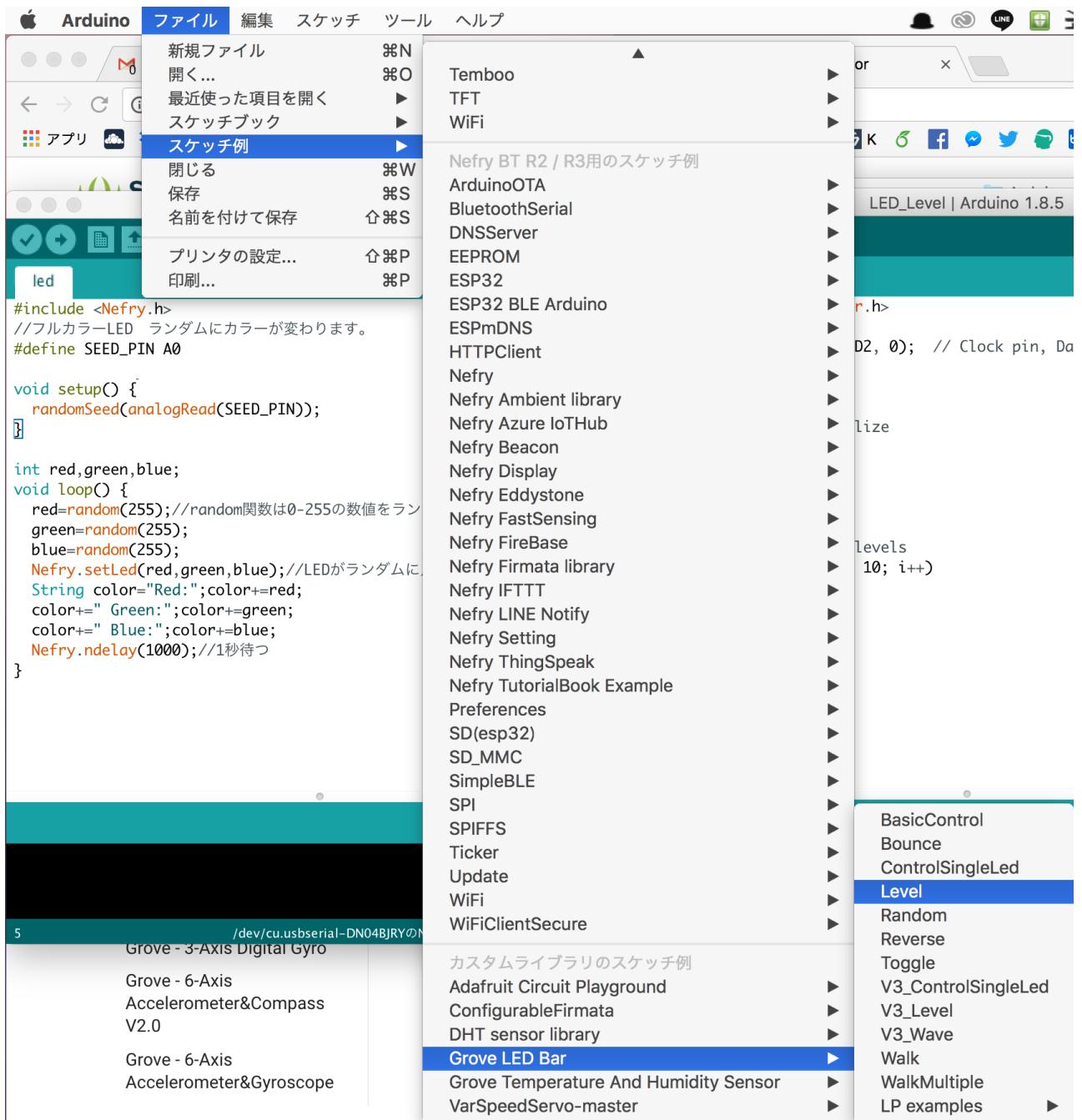
    if (isnan(h) || isnan(t) || isnan(f)) {
        NeFry.println("Failed to read from DHT sensor!"); // SerialをNeFryへ変更
        return;
    }

    float hif = dht.computeHeatIndex(f, h);
    float hic = dht.computeHeatIndex(t, h, false);
    NeFry.print("Humidity: ");      // SerialをNeFryへ変更
    NeFry.print(h);                // SerialをNeFryへ変更
    NeFry.print(" %\t");           // SerialをNeFryへ変更
    NeFry.print("Temperature: ");   // SerialをNeFryへ変更
    NeFry.print(t);                // SerialをNeFryへ変更
    NeFry.print(" *C ");           // SerialをNeFryへ変更
    NeFry.print(f);                // SerialをNeFryへ変更
    NeFry.print(" *F\t");          // SerialをNeFryへ変更
    NeFry.print("Heat index: ");    // SerialをNeFryへ変更
    NeFry.print(hic);              // SerialをNeFryへ変更
    NeFry.print(" *C ");           // SerialをNeFryへ変更
    NeFry.print(hif);              // SerialをNeFryへ変更
    NeFry.println(" *F");          // SerialをNeFryへ変更
}
```

## LEDbarセンサ

1. ライブラリ [https://github.com/Seeed-Studio/Grove\\_LED\\_Bar](https://github.com/Seeed-Studio/Grove_LED_Bar) をダウンロード
2. Arduino IDEを起動し、スケッチ > ライブラリをインクルード > ZIP形式のライブラリをインストール

### 3. スケッチ例 > Grove LED Bar > Levelをクリック



### LEDbarセンサ

#### 4. 下記プログラムを実行

```
#include <Grove_LED_Bar.h>
//Clock pin, Data pin, Orientation
Grove_LED_Bar bar(D3, D2, 0);

void setup()
{
    // nothing to initialize
    bar.begin();
}
```

```
void loop()
{
    // Walk through the levels
    for (int i = 0; i <= 10; i++)
    {
        bar.setLevel(i);
        delay(100);
    }
}
```