

# Nefry(ESP32) 說明書

---

V1.0.1

Library 1.0.3

2017/08/29

## 目次

---

<b>Nefryの仕様</b>	<b>3</b>
動作確認環境	3
ハードウェアの制約	3
Nefryライブラリーの仕様	3
ソフトウェアの仕様	3
電源について	3
ディスプレイについて	3
<b>Nefryの動作</b>	<b>4</b>
Nefryライブラリとは	4
Nefryライブラリのプログラムの動作について	5
書き込みモードとは？	6
LEDで見る動作状態	6
ディスプレイの扱いについて	6
Nefry Web Config	6
Wi-Fiの設定	7
Nefryの各種設定	8
プログラム書き込み	8
コンソール	9
<b>プログラム開発環境について</b>	<b>9</b>
Arduinoとは？	9
Arduino IDEの準備	9
Arduino IDEの設定	10
プログラム保存場所について	11
・ 方法	11
<b>プログラムの書き込み</b>	<b>11</b>
ローカルアップデート	12
<b>Nefryの機能</b>	<b>12</b>
スイッチ	12
フルカラーLED	13
Grove	13
各入出力ピンの機能	14
<b>Arduino言語</b>	<b>15</b>
SPI	15
I2C	15

デジタル入出力	15
アナログ入力	16
Pwm出力	16
<b>関連サイト</b>	<b>17</b>
<b>更新履歴</b>	<b>17</b>

このたびはNefryに興味を持っていただきありがとうございます。

NefryはMilkcocoaやmyThings、IFTTTなど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モジュールになっています。インターネットに接続する部分をNefryが請け負うので、インターネットへの接続するためのプログラムを書く時間をとられることなく上記のようなサービスを利用してあなたのアイデアを実現します！

## Nefryの仕様

---

### 動作確認環境

---

Windows10

Arduino IDE 1.8.2

Nefryライブラリー 0.9.0

Macでの動作はこちらでは保証できません。

### ハードウェアの制約

---

ESP-ROOM-32の2,5,6,7,8,9,10,11,12,15ピンは未配線のため使用することができません。

(0ピンは起動時ブートモード切替、4ピンはSWとして使用、16ピンはフルカラーLEDの制御ピンとして利用)

SSIDに日本語を使っているものは接続できません。

D0 D1はディスプレイとの通信用としてI2Cとして使用しています、I2Cのセンサであれば問題ありませんが、それ以外のセンサをつなぐとうまく動作しない可能性があります。

### Nefryライブラリーの仕様

---

webserverを使うことができません。

### ソフトウェアの仕様

---

ESP32 Arduino(<https://github.com/espressif/arduino-esp32>)をベースに開発しているため、そのバージョンに対応した機能のみ使用可能です。

Nefryライブラリーについては、リリースノートにてどの部分までマージしているか記載しているのでそちらをご覧ください。

### 電源について

---

NefryBTで採用しているESP32というモジュールは大量の電流を消費します。PCやバッテリーなどに 重大な影響をすることがあります。PCなどに接続する場合は、バスパワーをサポートするUSBハブなどを經由して接続することをお勧めします。基本的に1A以上のAC-USBの変換アダプタで接続することをお勧めします。

例:ELECOMのU3H-A408Sなど

### ディスプレイについて

---

液晶の扱いに注意してください。液晶が割れてしまうと、表示がおかしくなる可能性がありますので注意して扱ってください。ディスプレイの破損での交換などのサポートはありません。

## Nefryの動作

### Nefryライブラリとは

Nefryライブラリとは、ハードウェアと組み合わせてインターネットに繋ぐことを簡単にするために作られたライブラリです。以下の図のような構成となっています。



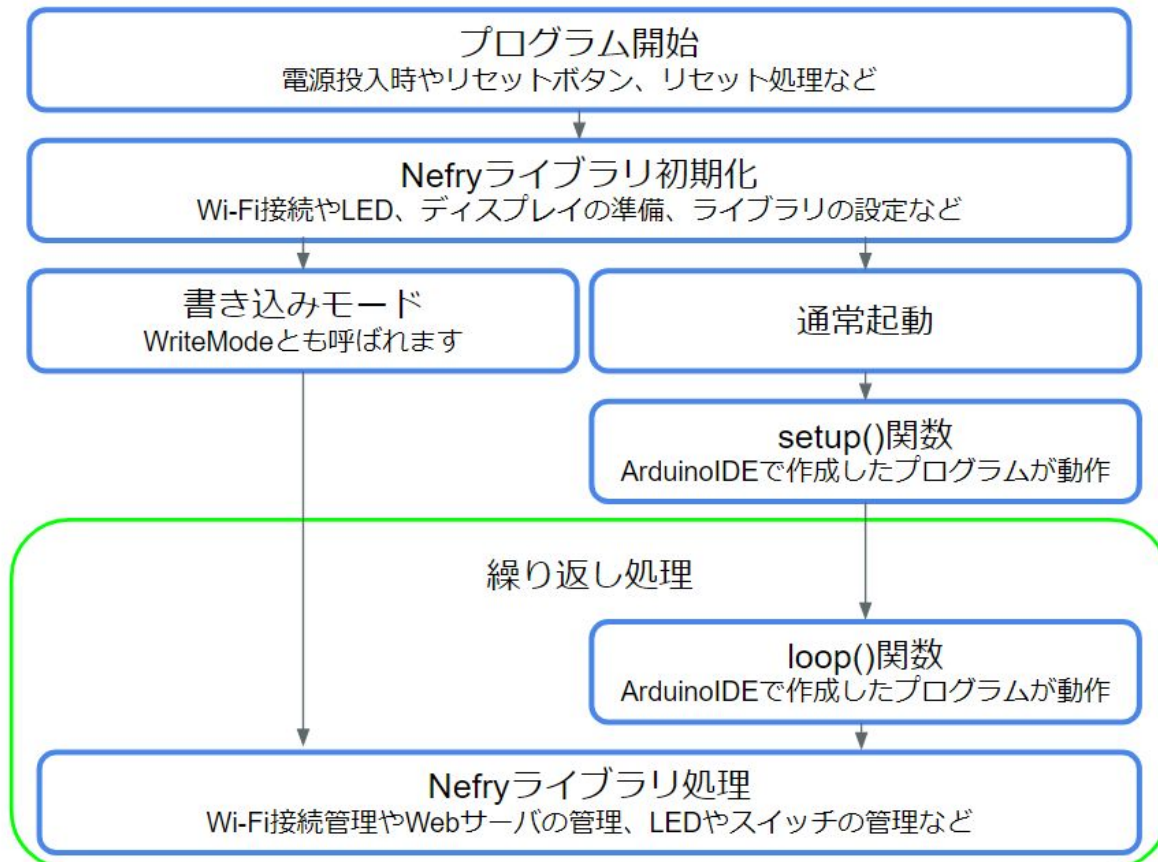
ESP-WROOM-32のソフトウェアはArduino ESP32(<https://github.com/espressif/arduino-esp32>)の機能をベースに開発しております。Nefryライブラリではそれをベースに後の章で紹介するWeb設定機能やアップデート機能、ハードウェア制御などを提供し、初めてでも簡単に作り始められるようにプログラムを作成しています。

その上でユーザプログラムが動作しています。そのためユーザのプログラム内でWi-Fiの接続などを意識せずに済むため、やりたいことだけに集中してプログラムを作ることができます。

この他にもNefryを便利に扱えるような関数を複数用意しております。そちらについてはこちら(<https://dotstud.io/docs/nefrybt/>)のサイトをご覧ください。またIFTTTやAzure IoT HubなどのWebサービスと簡単に接続できるようにライブラリも同封されてますのでぜひ試してみてください。

## Nefryライブラリのプログラムの動作について

NefryライブラリはArduinoのユーザプログラムが動作する前に動作を開始することにより、ユーザがWi-Fiやモジュールの設定を意識せずに簡単にプログラムを書くことができます。



電源投入時やリセットボタンを押した時などのプログラム開始後、Nefryライブラリの初期化コードが実行されます。このタイミングでWi-Fi接続、ディスプレイやLEDの準備などを実行します。この処理は条件にもよりますが、30秒から1分程度掛かります。

その後書き込みモード(WriteMode)か通常起動のどちらかに分岐します。その分岐条件は、書き込みモードでリセットか、Nefryライブラリ初期化中にNefryについているスイッチをおすことで書き込みモードに入ります。書き込みモードについての詳細は次で解説します。

通常起動した場合、ArduinoIDEでユーザが作成したsetup()関数がまず実行されます。それが完了した後、loop()関数が呼び出されます。

その後Nefryライブラリの処理が走り、その後再度loop()処理が呼ばれる流れとなっています。繰り返し処理の中NefryライブラリではWi-FiやWebサーバの管理などが行われています。

これまでで解説した内容がNefryライブラリのプログラム動作となります。

## 書き込みモードとは？

Nefryではオンライン書き込みによりプログラムを書き換えることができます。オンライン書き込みではネットワークを経由してプログラムを書き込みますので、Nefryでユーザプログラムが動いているとそのプログラムの書き込みに失敗する可能性が高くなります。そのため安全にプログラムを更新できるようにプログラム書き込みモードを用意しています。

・プログラム書き込みモードへの移行方法

- 緑色が点灯している間にResetボタンの反対にあるSWボタンを長押ししていただき、LEDが赤色点灯になってから手をはなしてください。

LEDが赤色点灯すれば、プログラム書き込みモードに移行できています。

- Nefry Web ConfigのSetup Moduleのページ内にWrite Modelに移行できるボタンがあり、そのボタンを押すと自動的にプログラム書き込みモードへ移行します。

## LEDで見る動作状態

電源を入れた直後やりセットボタンを押した際にはNefryのプログラムがWi-Fi接続、webserver開始準備などの準備をします。少々時間が掛かるためNefryについているLEDで動作確認できるようになっています。

起動完了後はユーザプログラムでLEDを制御できます。

Nefryに付いているLEDの状態	Nefryの動作状態
緑色点灯	起動準備中
緑色点滅	Wi-Fi接続実行中
青色点滅	Wi-Fi接続失敗
水色点灯	起動完了
赤色点灯	プログラム書き込みモード移行

## ディスプレイの扱いについて

NefryBTではディスプレイを搭載する事ができます。そのディスプレイには、起動時Nefryの情報を表示する事が出来ます。

ディスプレイとNefry間はI2Cでの接続となるため起動時にアドレス(0x3c)での通信が行われます。ディスプレイを接続している場合、ディスプレイについている抵抗により10kのプルアップ抵抗がかかります。

ディスプレイの詳しい使い方はこちら(<https://dotstud.io/docs/nefrybt/>)のサイトをご覧ください。

## Nefry Web Config

Nefry Web ConfigとはNefryで必要になるWi-Fiの設定やNefryの各種設定、プログラム書き込み、コンソール機能を備えたWebサイトです。

Nefryが出力しているWi-Fiに接続すると自動的にNefry Web Configが表示されるCaptive Portal機能が追加されています。この機能により簡単に扱えるようになっています。

上手く動作しない場合はNefryのWi-Fiに接続後、<http://192.168.4.1> に接続してください。スマートフォンなどでは接続に失敗することがあります。その際には機内モードにしたのちWi-Fiのみ有効にしてもう一度接続してみてください。



(Nefry Web Configのトップページ)

開発バージョンにより画面の変更などがある可能性があります。

## Wi-Fiの設定

Wi-Fiの接続の設定ができます。SSIDは選択して選ぶこともできます。

Wi-Fiは5つまで記憶することができ、起動中のネット接続時にもっとも電波が強いものに接続されます。5つ以上の保存をしようとするともっとも古いデータが削除され、新しいデータが追加されます。

2.4GHzのWi-Fiに接続できます。パスワードがない場合は入力せずSaveをしてください。

1. waminet (-27)
2. W04\_149D097C5BD6 (-39)
3. W03\_608334CF10C4 (-54)



## Nefryの各種設定

ここでNefryの環境変数を設定することができます。文字列と数値を入れるところがそれぞれ10個あります。この値はプログラムで取得、変更することができます。

String0-9は127文字のデータを保存することができます。

value10-19はlongの範囲の数値を保存することができます。

### 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。

- 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公式サイトをご覧ください。

String 0	<input type="text"/>
String 1	<input type="text"/>
String 2	<input type="text"/>
Value 10	<input type="text" value="0"/>
Value 11	<input type="text" value="0"/>
Value 12	<input type="text" value="0"/>

[WriteMode](#)  
[Back to top](#)

## プログラム書き込み

Nefryのプログラムを更新します。Nefryにはローカルからのアップロードで更新するローカルアップロードがあります。具体的な方法は別の章で解説します。

### Upload Sketch

Upload a binary file of sketch.

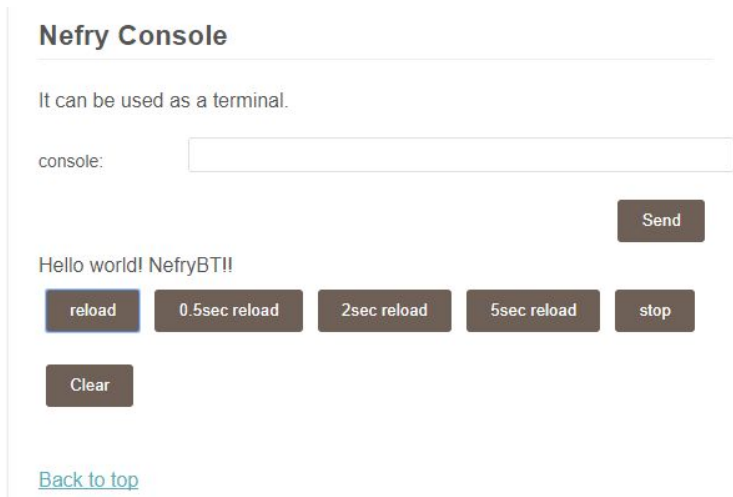
arduino.bin

[Back to top](#)

## コンソール

---

Nefryではデバッグや動作状態を確認することのできるコンソールがあります。入力も可能なため、入力データにより動作を変化させることもできます。



## プログラム開発環境について

---

NefryはArduino IDEでプログラミングをします。この章ではArduino IDEの環境構築について解説していきます。

### Arduinoとは？

Arduinoとは、電子工作に興味を持った方やハードウェアを使って簡単に作品を作りたい方におすすめのものです。

Arduinoを使うメリットは、ハードウェアの難解なところを簡単なArduino言語と呼ばれるもので開発できるようになることです。

### Arduino IDEの準備

---

Arduino IDEは現在も開発されており、日々改善されております。基本的に最新のバージョンでも動作するはずですが、大幅な仕様変更により動作しなくなることも考えられます。

現状こちらで動作を確認しているバージョンはArduinoIDE 1.8.2になります。それ以外で動かない場合はこちらのバージョンに合わせてみると動作するはずです。

ダウンロードサイト<https://www.arduino.cc/en/Main/Software>

Windowsの方はWindows Installerをダウンロードしてインストールするのがおすすめです。

Macの方はMAC OS Xをダウンロード、インストールをよろしくお願いします。

## Arduino IDEの設定

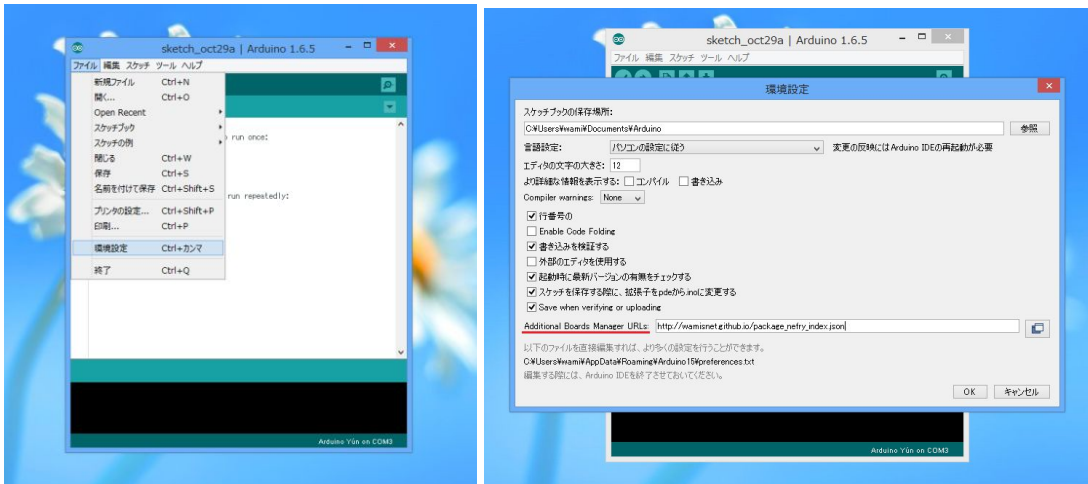
Arduino IDEのインストールが終了したのち、Arduino IDEでNefryのプログラムを作るように準備しましょう。

1. ファイル→環境設定を開きます。環境設定が開いたらAdditional Boards Manager URLsに次のURLを入力します。

安定版(基本はこちらを選択) : [http://nefry.studio/package\\_nefrybt\\_index.json](http://nefry.studio/package_nefrybt_index.json)

開発版 : [http://nefry.studio/develop/package\\_devnefrybt\\_index.json](http://nefry.studio/develop/package_devnefrybt_index.json)

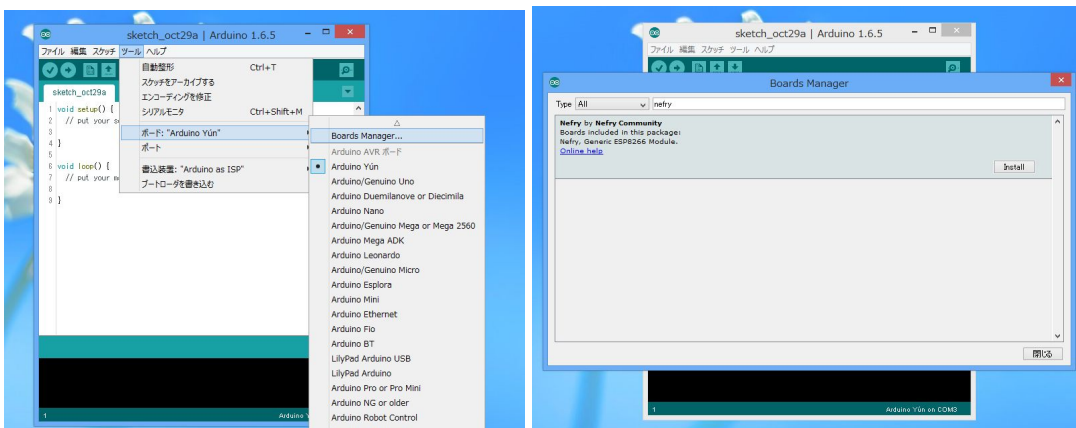
入力が終わったらOKを押して戻ります。



2. ツール→マイコンボード→ボードマネージャーを開きます。

上の入力欄に"Nefry"と入力していただくと、Additional boards manager URLsで追加したURLを検索し、Nefryと出るので右下にある"Install"をクリックしてインストール完了までしばらくお待ちください。

インストールが完了すれば、NefryをArduino IDEで開発できるようになります。



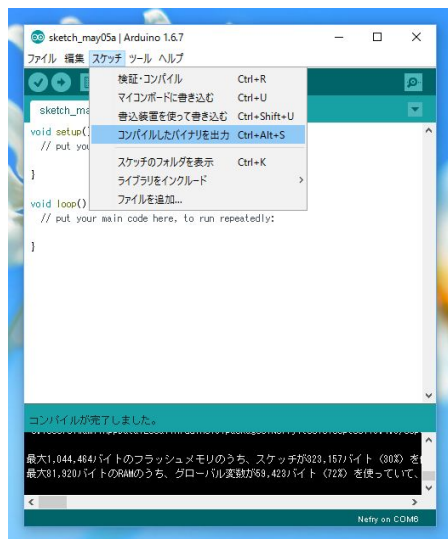
## プログラム保存場所について

この項目ではNefryに書き込むファイルの保存場所について解説します。

Arduino IDEのバージョンによって異なります。Nefryの推奨はArduino IDE 1.8.2以上です。

### ・方法

Arduinoのプログラムファイルを保存したフォルダに、“スケッチ→コンパイルしたバイナリを出力”をクリックするとNefryに書き込むファイルが生成されます。



## プログラムの書き込み

Nefryに新しいプログラムを書く方法は3つあります。それぞれの特徴は以下になっています。

- ~~Webserverからダウンロードして更新するオンラインアップロード~~ **未対応**
- ローカルからのアップロードで更新するローカルアップロード
- シリアル変換を使いプログラムを更新するハードウェア書き込み

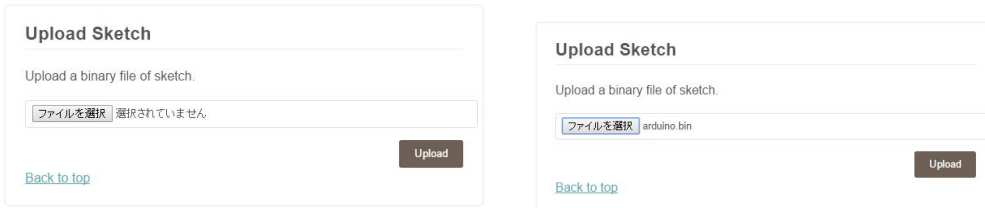
PCに直接接続せずに使う際に使い分けとして、ローカル環境（個人での開発）ではローカルアップロードがおすすめです。複数のNefryにプログラムを配布する際におすすめなのがオンラインアップデートになります。

## ローカルアップデート

---

ローカルアップデートはインターネットへの接続は不要です。Nefryに直接接続してプログラムをアップロードすることでプログラムを更新します。

Nefry Web ConfigのUpload Sketchにある“ファイルを選択”（Chrome）や“参照”（Edge）などファイルを選択する部分から（arduino.binファイル保存場所については、[プログラム保存場所の変更](#)の項目で解説しています。）arduino.binという名前のファイルを探し、アップロードすることでプログラムを書きかえられます。



## Nefryの機能

---

### スイッチ

Nefryに付いている押しボタンです。Nefry.readSW()と呼び出すことでプログラム中にて使用することができます。

サンプルプログラム SWを押した時にLEDが青色に点灯する

```
#include <Nefry.h>

void setup() {
}

void loop() {
  //SWを押した時にLEDが青色に点灯する
  if(Nefry.readSW()){
    //押した時 青色になる（r,g,b）の順で色を指定できる
    Nefry.setLed (0,0,255);
    Nefry.println("Push SW");
  }else{
    //離れた時
    Nefry.setLed(0,255,255);
  }
  delay(100);
}
```

### フルカラーLED

フルカラーLEDを接続しているので、これを使用して簡単な動作状況の表示が可能です。ほかにもプログラミングが可能ですので、これを使って遊ぶこともできます。

サンプルプログラム ランダムにLEDの色が変わります！

```
#include <Nefry.h>
```

```
//フルカラーLED ランダムにカラーが変わります。
```

```
void setup() {
```

```
  Nefry.println("フルカラーLED！");
```

```
  randomSeed(analogRead(A0));
```

```
}
```

```
void loop() {
```

```
  Nefry.setLed(random(255),random(255),random(255));//ランダムに点灯します。
```

```
  delay(500);
```

```
}
```

## Grove

---

それぞれ別の機能を持つgroveのソケットを持っています。電源は3.3vを供給します。  
アナログ入出力,デジタル入出力,I2Cの3つの機能があります。

## 各入出力ピンの機能

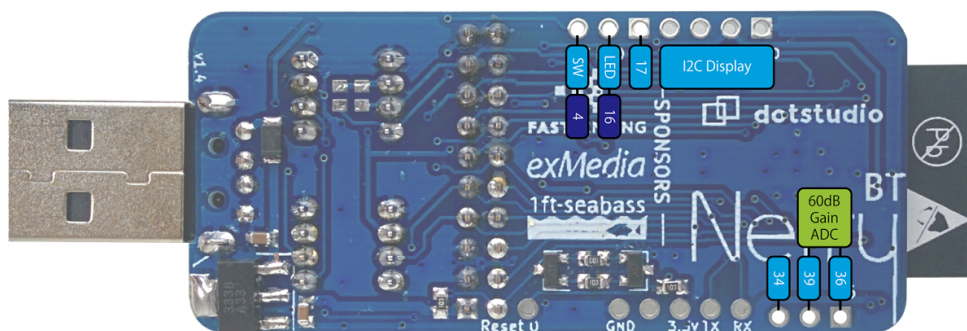
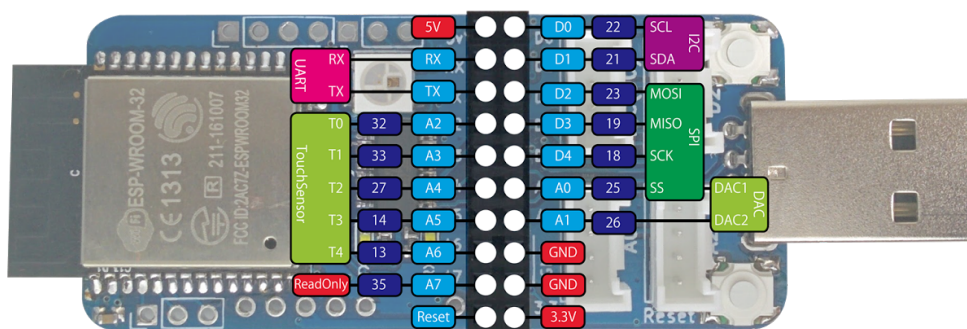
下の図のピン番号は、プログラムはこちらで提供するArduino IDEで書き込み時に使用可能です。

### 一列目

NefryBTピン	3.3v	GND	GND	A1	A0	D4	D3	D2	D1	D0
機能	3.3v	GND	GND	DAC2	DAC1 SPI CS	SPI SCK	SPI MISO	SPI MOSI	I2C SDA	I2C SCL
ESP32ピン	3.3v	GND	GND	26	25	18	19	23	21	22

### 二列目

NefryBTピン	Reset	A7	A6	A5	A4	A3	A2	TX	RX	5V
機能	Reset	Read only	Touch T4	Touch T3	Touch T2	Touch T1	Touch T0	UART	UART	5v
ESP32ピン	EN	35	13	14	27	33	32	TX	RX	



- 5V ... 電源
- D0 ... NefryBTのピン指定
- 22 ... ESP-WROOM-32のピン指定

## Arduino言語

---

### SPI

---

SPI通信は、D2(CS),D3(MOSI),D4(MISO),D5(CLK)ピンを使用します。

### I2C

---

I2C通信は、D0(SCL),D1(SDA)ピンを使用します。

## デジタル入出力

---

デジタル入出力対応ピンについてはピンマップをご覧ください。HIGHのとき3.3vを出力し、LOWの時0vを出力します。

サンプルプログラム

- SWを押した時にA2にLEDを差すと点灯する
- D4がスイッチなどでLOWになったときにNefryのLEDが青色に光る

```
#include <Nefry.h>
void setup() {
  pinMode(A2,OUTPUT);//出力モード切り替え
  pinMode(D4,INPUT_PULLUP);//入力モード切り替え
}
void loop() {
  //SWを押した時にD5にLEDを差すと点灯する
  if(Nefry.push_SW()){
    //押した時
    digitalWrite(A2,HIGH);
    Nefry.println("Push SW");
  }else{
    //離れた時
    digitalWrite(A2,LOW);
  }

  //D4がLOWになったときにNefryのLEDが青色に光る
  if(digitalRead(D4)==LOW){
    //青色になる (r,g,b) の順で色を指定できる
    Nefry.setLed(0,0,255);
    Nefry.println("D4 LOW");
  }
}
```



```

}else{
  //緑色になる
  Nefry.setLed(0,255,0);
}
delay(100);
}

```

## アナログ入力

---

アナログ入力のピンについてはピンマップをご覧ください。

サンプルプログラム アナログ入力の数値をNefry Web Config のコンソールに表示する

```

#include <Nefry.h>
int sensorValue;
void setup() {
}
void loop() {
  // read the analog in value:
  sensorValue = analogRead(A0);
  Nefry.print("sensor = " );
  Nefry.println(sensorValue);
  delay(1000);
}

```

## Pwm出力

---

Pwm出力に対応したピンはデジタル入出力ピンすべて（D0-D5）です。3.3vを256段階で変化させていきます。

サンプルプログラム 徐々にD5ピンに接続されたLEDの明るさを変えていく

```

void setup() {
}
int led_value ;
void loop() {
  for (led_value = 0; led_value <= 255; led_value += 5 ) {
    analogWrite(D5, led_value );
    delay( 30 );
  }
  for (led_value = 255; led_value >= 0; led_value -= 5 ) {
    analogWrite(D5, led_value );
    delay( 30 );
  }
}

```

```
}  
}
```

## 関連サイト

---

Dotstudio

- Nefry BTとIFTTTでスイッチを押したらLINEを送る仕組みを作ってみよう( <https://dotstud.io/blog/nefry-ifttt-push-line/> )

## 更新履歴

---

v1.0 2017/8/1 初版

v1.1 2017/8/29 サンプルプログラムの修正