

Nefry 說明書

V1.0

2016/03/28

目次

Nefry の仕様.....	2
動作確認環境	2
ハードウェアの制約	2
Nefry ライブラリーの仕様.....	3
ソフトウェアの仕様	3
Nefry の動作.....	3
LED で見る動作状態	3
プログラム書き込みモードとは?	3
Nefry Web Config	4
Wi-Fi の設定	4
Nefry の各種設定.....	4
プログラム書き込み	5
コンソール	6
プログラム開発環境について	6
Arduino とは?	6
Arduino IDE の準備	6
Arduino IDE の設定	7
プログラム保存場所の変更.....	8
• 1.6.5 以下のバージョン	8
• 1.6.6 以上のバージョン	9
簡易設定ツールについて.....	10
プログラムの書き込み.....	11
オンラインアップデート.....	11
ローカルアップデート	12
ハードウェア書き込み	12
Nefry の機能.....	13
スイッチ	13
フルカラーLED	14

Grove	15
各入出力ピンの機能	15
スリープ機能	16
Arduino 言語.....	16
SPI.....	16
I2C	16
ディジタル入出力.....	18
アナログ入力	19
Pwm 出力	20

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 1.6.5

Nefry board 1.1.0

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-02 の 0,4,16 ピンは予約済みのため使用することができません。

（0 ピンは起動時とフルカラーLED にて使用、4 ピンは SD カードと SW にて使用、16 ピンはリセットピンとして利用）

SSID に日本語を使っているものは接続できません。

SD カードと SW は同時利用することができません。

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

esp8266 Arduino(<https://github.com/esp8266/Arduino>)こちらで開発している Staging version 2.0 rc1 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 の動作

LED で見る動作状態

電源を入れた直後やリセットボタンを押した時、プログラムリセットの際には 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

プログラム書き込みモードとは？

Nefry では別の章で解説しますが、オンライン書き込みによりプログラムを書き換えます。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プログラムの書き込みに失敗する可能性が高いです。そのため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

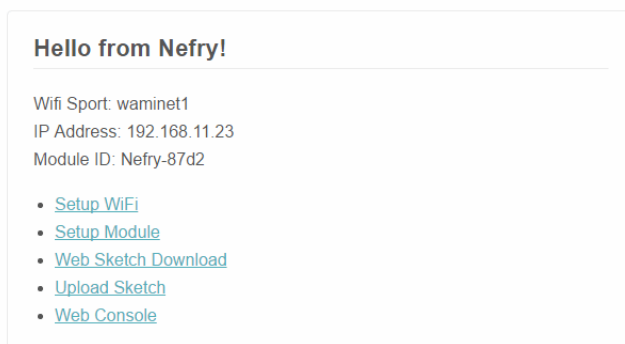
緑色が点灯している間に Reset ボタンの反対にある SW ボタンを長押ししていただき、LED が赤色点灯になってから手をはなしてください。

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

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 Web Config には Nefry の Wi-Fi に接続し、<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。

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。

Nefry Wifi Set	
SSID:	<input type="text"/>
PASS:	<input type="text"/>
Back to top <input type="button" value="Save"/>	

1. waminet1 (-52)* 2. 000a79d22db7 (-93)*	
--	--

Nefry Wifi Set	
SSID:	<div><div></div><div>waminet1</div></div>
PASS:	<div><div></div><div>000a79d22db7</div></div>
Back to top <input type="button" value="Save"/>	

1. waminet1 (-52)* 2. 000a79d22db7 (-93)*	
--	--

Nefry の各種設定

ここで Nefry に関するさまざまな設定ができます。

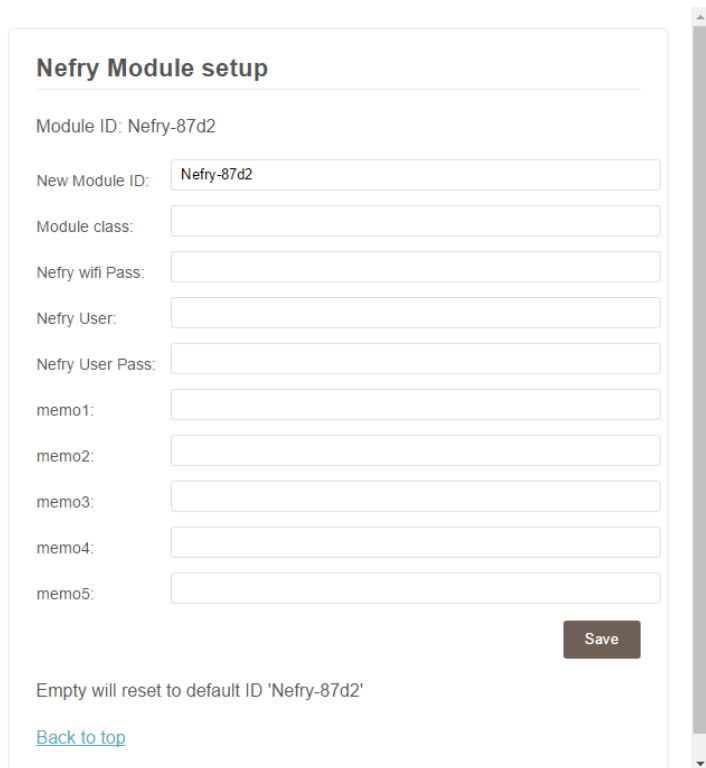
Module ID は Nefry が出す Wi-Fi の SSID にもなります。

Module class は Nefry の管理をするために class を使い複数の端末を制御します。

Nefry wifi Pass は Nefry が出す Wi-Fi に Password をつけることができます。8 文字以上です。

Nefry User、Nefry User pass は Nefry の認証に使われます。

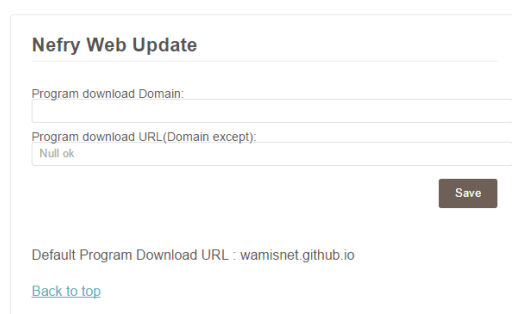
Memo1-3 は 127 文字、Memo4-5 は 63 文字のデータを保存することができます。



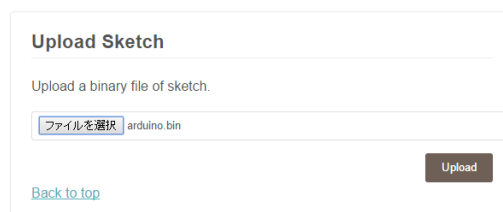
The screenshot shows a web form titled "Nefry Module setup". It contains several input fields: "Module ID" (pre-filled with "Nefry-87d2"), "New Module ID" (pre-filled with "Nefry-87d2"), "Module class", "Nefry wifi Pass", "Nefry User", "Nefry User Pass", and five "memo" fields (memo1 through memo5). A "Save" button is located at the bottom right of the form. Below the form, there is a note: "Empty will reset to default ID 'Nefry-87d2'" and a link "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 Webserver からダウンロードして更新するオンラインアップデートとローカルからのアップデートで更新するローカルアップデートがあります。具体的な方法は別の章で解説します。



The screenshot shows a web form titled "Nefry Web Update". It contains two input fields: "Program download Domain:" and "Program download URL(Domain except):". Below the second field, it says "Null ok". A "Save" button is located at the bottom right of the form. Below the form, there is a note: "Default Program Download URL : wamisnet.github.io" and a link "Back to top".



The screenshot shows a web form titled "Upload Sketch". It contains a text input field with a file selection button "ファイルを選択" and the text "arduino.bin". An "Upload" button is located at the bottom right of the form. Below the form, there is a link "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。

Nefry Console

It can be used as a terminal.

console:

Send

welcome nefry
welcome nefry

[Back to top](#)

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。しかしながら新しいバージョンによっては Nefry のライブラリーが対応していないものもあるので、今回は **1.6.5** というバージョンの Arduino IDE をインストールしてください。

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

Arduino 1.6.x, 1.5.x BETA

These packages are no longer supported by the development team.

1.6.6	Windows Windows Installer	MAC OS X	Linux 32 Bit Linux 64 Bit	Source code on Github
1.6.5	Windows Windows Installer	MAC OS X	Linux 32 Bit Linux 64 Bit	Source code on Github
1.6.4	Windows Windows Installer	MAC OS X	Linux 32 Bit Linux 64 Bit	Source code on Github
1.6.3	Windows Windows Installer	MAC OS X	Linux 32 Bit Linux 64 Bit	Source code on Github
1.6.2	Windows Windows Installer	MAC OS X	Linux 32 Bit Linux 64 Bit	Source code on Github
1.6.1	Windows Windows Installer	MAC OS X MAC OS X Java 7+	Linux 32 Bit Linux 64 Bit	Source code on Github
1.6.0	Windows Windows Installer	MAC OS X MAC OS X Java 7	Linux 32 Bit Linux 64 Bit	Source code on Github

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。

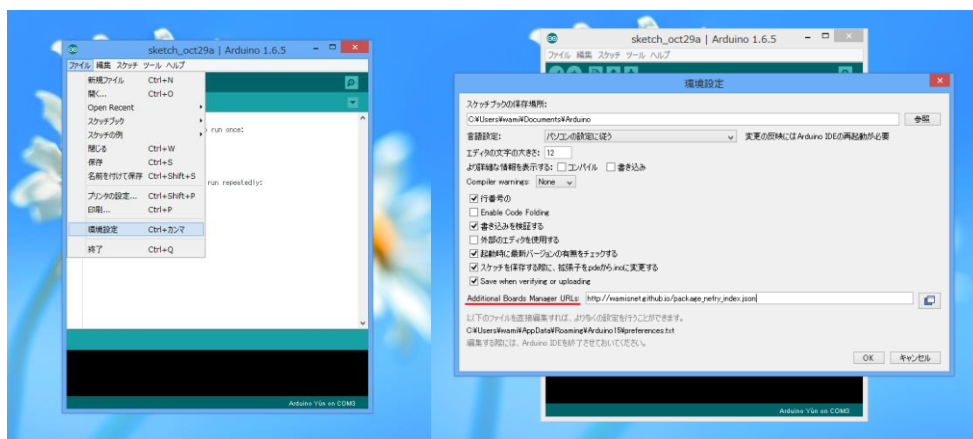
Arduino IDE の設定

Arduino IDE のインストールが終了したのち、NefryはArduino IDEに元々入っていないので追加作業をします。これを行うことでNefryのプログラムを書くことができるようになります。

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。

http://wamisnet.github.io/package_nefry_index.json

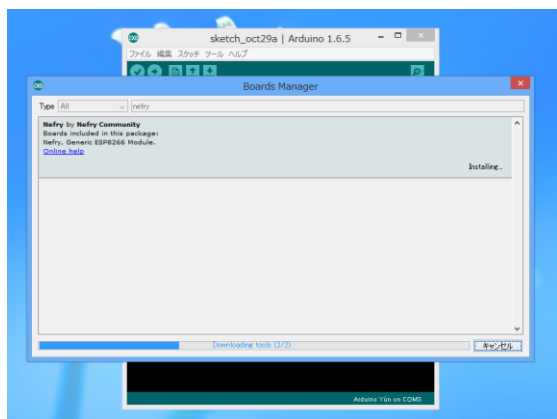
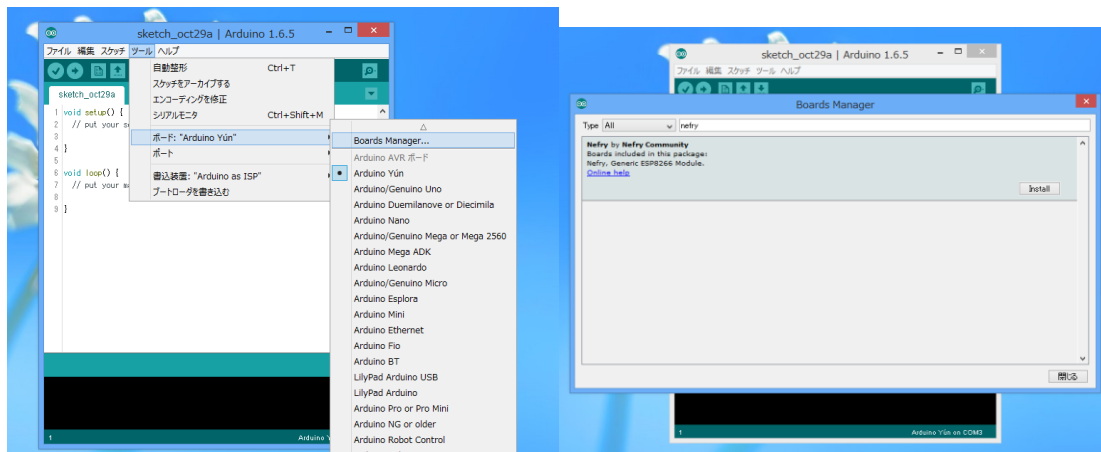
入力が終わったらOKを押して戻ります。



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の変更

この項目では Nefry に書き込むファイルの保存場所を変更します。

変更しないでつかうこともできますが、ファイルが作られる場所が非常にわかりにくいいため必ず変更作業をしてください。

Arduino IDE のバージョンによって異なります。Nefry の推奨は 1.6.5 です。

• 1.6.5 以下のバージョン

• Windows

Windows の場合には簡易設定ツールがあります。こちらより[ダウンロード](#)

(<http://wamisnet.github.io/Nefry-1.6.5.bat>)

手動設定方法

Arduino IDE を開き、ファイル→環境設定を開きます。環境設定が開いたらファイルへのリンクをクリックしてください。（写真にある赤線の部分）

Preferences ファイルをメモ帳などのテキストエディタで編集します。次の一行を追加します。

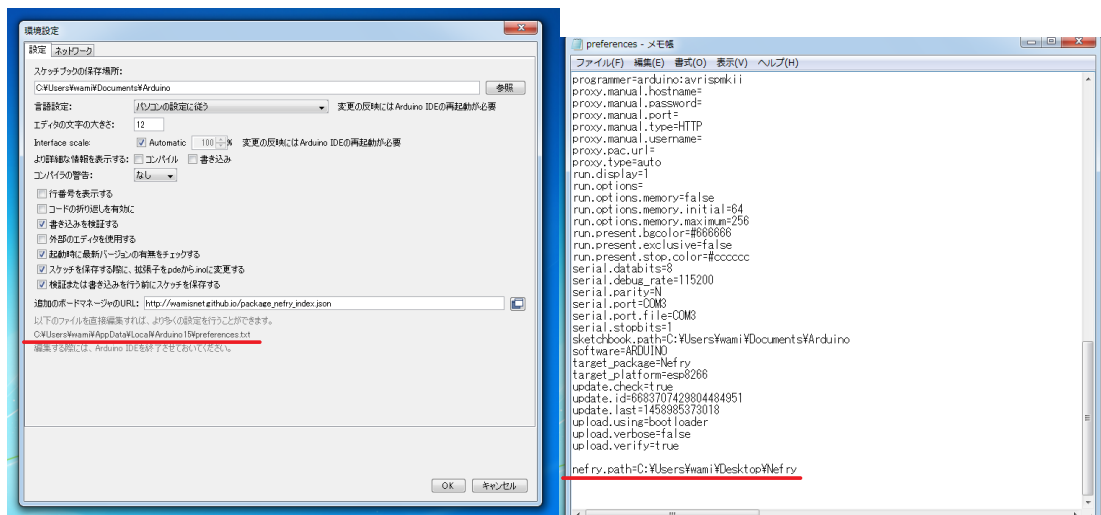
例：nefry.path=C:\Users\wami\Desktop\Nefry 記入法：nefry.path=保存フォルダ名

ここで指定したフォルダ名の座標に Nefry に書き込むファイルが出来上がります。

もう一つ設定があります。

Preferences ファイルが入っているフォルダから辿り、packages→Nefry→hardware→esp8266→バージョン（例：1.1.0）の中にある platform ファイルをメモ帳などのテキストエディタで編集します。メモ帳などテキストエディタの機能である置換を使い、{build.path}/arduino.bin を {nefry.path}/arduino.bin に置換してください。

以上で設定は終わりです。



• Mac

基本的には windows の手動設定と同じです。

ファイルの座標設定が分かりにくいので、一例をあげておきます。

例) /Users/ユーザ名/Documents/Nefry …ドキュメントにファイルを保存します

/User/ユーザ名/Desktop/Nefry …デスクトップにファイルを保存します

• 1.6.6 以上のバージョン

以前に 1.6.5 以前の Arduino IDE をインストールしていた場合、Nefry ファイルの削除が必要となります。

• Windows

Windows の場合には簡易設定ツールがあります。こちらより[ダウンロード](http://wamisnet.github.io/Nefry1.6.6-.bat)
(<http://wamisnet.github.io/Nefry1.6.6-.bat>)

手動設定方法

Arduino IDE を開き、ファイル→環境設定を開きます。環境設定が開いたらファイルへのリンクをクリックしてください。

Preferences ファイルをメモ帳などのテキストエディタで編集します。次の一行を追加します。

例：nefry.path=C:\Users\wami\Desktop\Nefry 記入法：nefry.path=保存フォルダ名

ここで指定したフォルダ名の座標に Nefry に書き込むファイルが出来上がります。

もう一つ設定があります。

Preferences ファイルが入っているフォルダから辿り、packages→Nefry→hardware→esp8266→バージョン（例：1.1.0）の中にある platform ファイルをメモ帳などのテキストエディタで編集します。メモ帳などテキストエディタの機能である置換を使い、{build.path}/arduino.bin を {nefry.path}/arduino.bin に置換してください。置換が終わりましたら、Preferences ファイルで追加した nefry.path で始まる一行を platform ファイルに追加してください。

以上で設定は終わりです。

• Mac

基本的には windows の手動設定と同じです。

ファイルの座標設定が分かりにくいので、一例をあげておきます。

例) /Users/ユーザ名/Documents/Nefry …ドキュメントにファイルを保存します

/User/ユーザ名/Desktop/Nefry …デスクトップにファイルを保存します

簡易設定ツールについて

簡易設定ツールとは Windows 向けに作られた、手動設定を自動化するためのものです。

ファイルの座標の指定をするだけで使うことができ、またファイルの保存先を変更したい際にはもう一度簡易設定ツールを使うことで変更が可能になっています。

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デート
- ローカルからのアップデートで更新するローカルアップデート
- 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

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デートがおすすめです。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。

オンラインアップデート、ローカルアップデートで上手くいかない時に、ハードウェア書き込みを試してみてください。

オンラインアップデート

オンラインアップデートには、インターネットに繋がっていないと動作しませんのでご注意ください。

ファイル名についてはプログラムをコンパイルするとできる（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使用します。

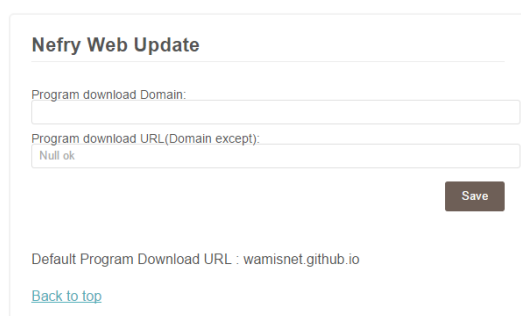
一例として wamisnet.github.io/arduino.bin（サーバー直下）にファイルがある場合

Program download Domain に wamisnet.github.io、Program download URL(Domain except)に無記入で send を押してください。

他の例として wamisnet.github.io/Nefry/arduino.bin にファイルがある場合

Program download Domain に wamisnet.github.io、Program download URL(Domain except)に Nefry で send を押してください。

エラーの時は Web ページで確認できますが、プログラム更新に成功した場合は Web ページでは確認できませんが、LED が水色点滅することで更新が完了したかどうか分かります。

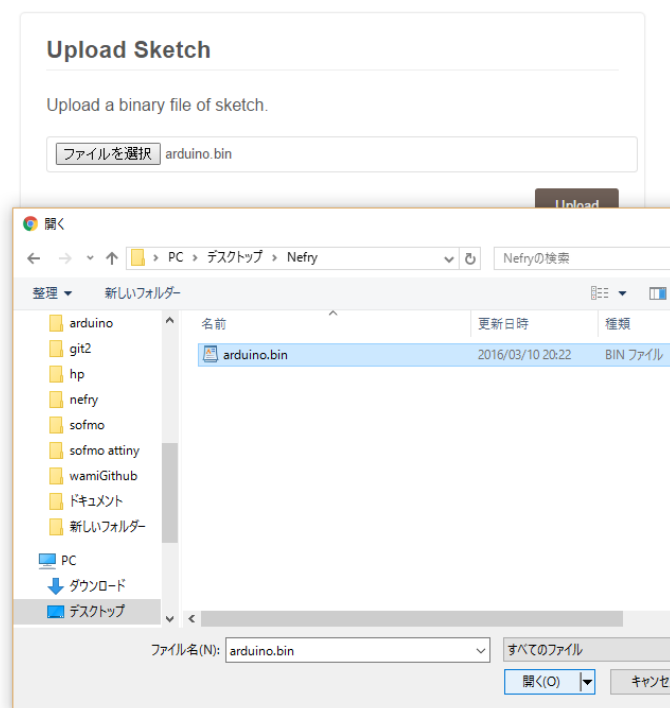


The screenshot shows a web form titled "Nefry Web Update". It contains two input fields: "Program download Domain:" and "Program download URL(Domain except):". The second field has "Null ok" written below it. A "Save" button is located to the right of the second field. At the bottom, it displays "Default Program Download URL : wamisnet.github.io" and a link "Back to top".

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。



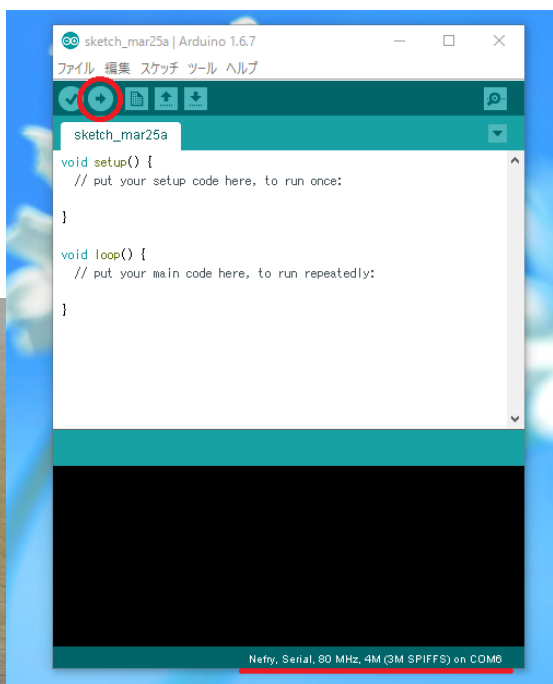
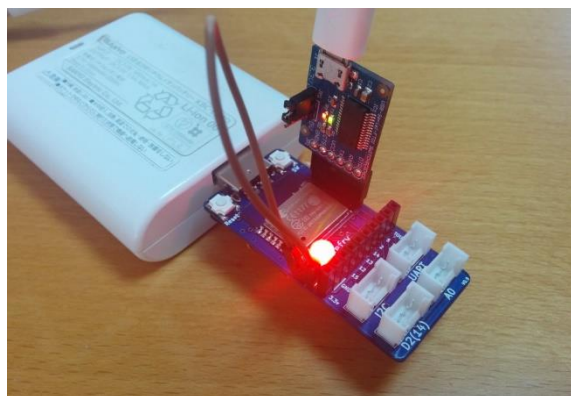
ハードウェア書き込み

ハードウェア書き込みについては、Switch Science さんで発売されている USB シリアル変換アダプタ（例：<https://www.switch-science.com/catalog/1032/>）などが必要になります。

注意：3.3V で必ず書き込みをしてください。

Nefry には 6 つの穴が開いていて、GND,3.3v,TX,RX を接続します。裏側にどのピンを接続するか印刷があるので確認してください。

1. Nefry の裏側に Write と書いてある 2 つの穴を繋いでください。（写真では茶色の線で繋いでいるところ）
2. Write の 2 つの穴を繋いだのちリセットボタンを押してください。
3. Arduino IDE で書き込みます。ボードが Nefry になっているか、シリアルポートが合っているかかくにんしてください。（写真では、赤丸が書き込みボタン、下にある赤線が Nefry の設定 環境によってはシリアルポート番号（com6）ではない可能性があります。）
4. 書き込みが始まったら Write の接続を解除してください。



Nefry の機能

スイッチ

Nefry に付いているプッシュ SW です。Nefry.pushSW()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 が赤色に点灯する

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

```

void setup() {
}

void loop() {
    //SW を押した時に LED が赤色に点灯する
    if(Nefry.push_SW()){
        //押した時
        //赤色になる（r,g,b）の順で色を指定できる
        Nefry.Nefry_LED(0,0,255);
        Nefry.println("Push SW");
    }else{
        //離れた時
        Nefry.Nefry_LED(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.Nefry_LED(random(255),random(255),random(255)); //ランダムに点灯します。
    delay(500);
}

```

Grove

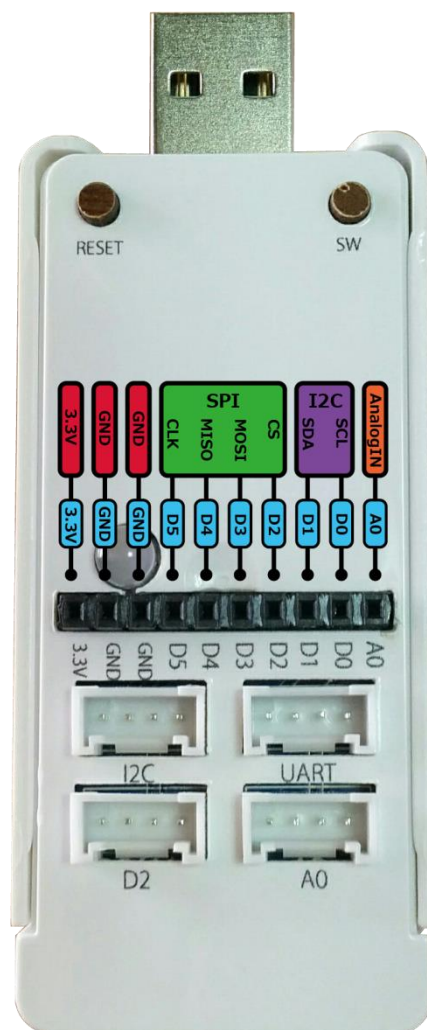
それぞれ別の機能を持つ grove のソケットを持っています。電源は 3.3v を供給します。

UART, アナログ入力, デジタル入出力, I2C の 4 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。

Nefry 入出力 ピン	3.3v	GND	D5	D4	D3	D2	D1	D0	A0
機能	3.3v	GND	SPI CLK	SPI MISO	SPI MOSI	SPI CS	I2C SDA	I2C SCL	Analog input
ESP8266 入 出力ピン	3.3v	GND	14	12	13	15	2	5	TOUT



スリープ機能

スリープモードを活用するととても低い消費電力になるため、バッテリーにて長期間の運用をすることができます。スリープモードの後はプログラムを初めからやり直します。

サンプルプログラム 5 秒青色の LED が点灯し、その後 30 秒スリープします。

```
#include <Nefry.h>

//Sleep モードの例 30 秒したらスリープから復帰します。

void setup() {

}

void loop() {
    Nefry.println("Sleep!");
    Nefry.Nefry_LED(0, 0, 255); //青色に点灯します。
    delay(5000);
    Nefry.Nefry_LED(0, 0, 0, 0); //消灯します。
    Nefry.sleep(30); //秒数で指定します。
}
```

Arduino 言語

SPI

SPI 通信は、D2(CS),D3(MOSI),D4(MISO),D5(CLK)ピンを使用します。プルアップ抵抗が入っています。

I2C

I2C 通信は D0(SCL),D1(SDA)ピンを使用します。プルアップ抵抗が入っています。

Wire.begin();で動作します。

サンプルプログラム DRV8830 ドライバーを使ってモータを回します。

/*DRV8830 ドライバーを使ってモータを回します。

*

* SDA:D1

```

* SCL:D0
*
* -----
* DRV8830 接続例 http://akizukidenshi.com/catalog/g/gK-06273/
* DRV8830---Nefry
*
* SCL---D0
* SDA---D1
* A1---GND
* A0---GND
* FAULTn---open
* GND---GND
* VCC---3.3v
* OUT1---motor
* ISENSE---open
* OUT2---motor
*
* /

```

```
#include <Wire.h>
```

```
#define ADDR 0x60
```

```
void setup() {
```

```
Wire.begin();
```

```
}
```

```
void start_motor(){
```

```
Wire.beginTransmission(ADDR);
```

```
Wire.write(0x00);
```

```
Wire.write(0xFD);
```

```

    Wire.endTransmission();
}

void stop_motor(){
    Wire.beginTransmission(ADDR);
    Wire.write(0x00);
    Wire.write(0x03);
    Wire.endTransmission();
}

void loop() {
    // モータを回転、停止するプログラム
    start_motor();
    delay(1000);
    stop_motor();
    delay(1000);
}

```

ディジタル入出力

ディジタル入出力はディジタル入出力ピンすべて（D0-D5）です。3.3v を出力します。

●注意：d5 ピン以外のピンにはプルアップ抵抗が入っています。

サンプルプログラム SW を押した時に D5 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(D5,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(D5,HIGH);
    Nefry.println("Push SW");
}else{
    //離れた時
    digitalWrite(D5,LOW);
}

//D4 が LOW になったときに Nefry の LED が青色に光る
if(digitalRead(D4)==LOW){
    //赤色になる (r,g,b) の順で色を指定できる
    Nefry.Nefry_LED(0,0,255);
    Nefry.println("D4 LOW");
}else{
    //緑色になる
    Nefry.Nefry_LED(0,255,255);
}
delay(100);
}

```

アナログ入力

アナログ入力は A0 ピンのみ対応です。分圧抵抗を入れているので 3.3v 入力をしてください。

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 );
  }
}

```