

LoRa IoT スターターキット スタートアップガイド

OPENWAVE.INC

LoRa IoT スターターキット スタートアップガイド

目次

1. 本書について	2
2. 全体像	2
3. 使用するもの	3
4. IoT サーバにアップロードするまでの操作手順.....	4
4.1. ThingSpeak の設定	4
4.2. LoRa プログラミングの前に.....	7
4.3. LoRa ゲートウェイのプログラミング.....	8
4.4. LoRa ノードのプログラミング	10
4.5. LoRa Mini と温度湿度センサーをつなぐ	11
4.6. 接続確認	12

1. 本書について

LoRa IoT スターターキットは、LoRa 通信を簡単に体験するためのキットです。本書では、LoRa IoT スターターキットを使って LoRa 通信の実施と、温度湿度センサーのデータを IoT サーバにアップロードするまでの操作方法について説明します。

2. 全体像

スターターキットに含まれるものを使用し、温度湿度センサーのデータを、外部の IoT サーバ ThingSpeak にアップロードします。

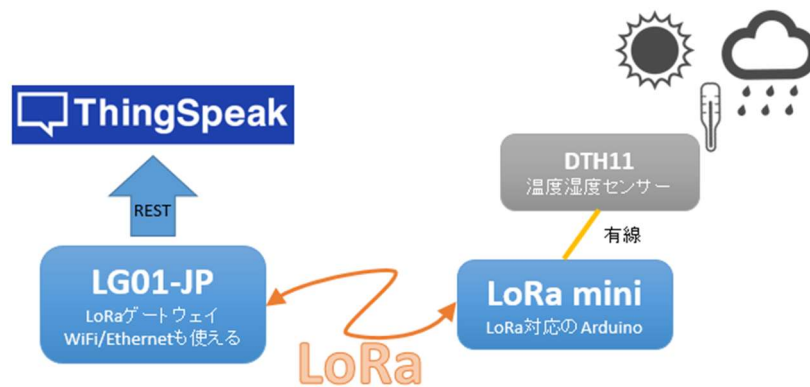


図 1.本書で実施する操作の全体構成



図 2.完成時の全体写真

3. 使用するもの

本手順では、下記のものを使用します。

- LoRa IoT スターターキット

- (O)LG01-P-JP
- LoRa Mini Dev-JP (以降 LoRa Mini)
- DHT11 (温度センサー)

- USB 接続のバッテリー

スマホの充電用バッテリー (USB 端子のもの) で構いません。無い場合は PC に接続による給電を使用します。

- PC (USB 端子のあるもの)

- Arduino IDE

前提として、LG01-JP の下記マニュアルに従い、設定がなされているものとします。

http://www.openwave.co.jp/LG01/LG01_LoRa_Gateway_Manual_JP.pdf

- サンプルプログラム (下記 GitHub より取得)

<https://github.com/openwave-co-jp/LoRaToThingSpeak>

- LoRaToThingSpeak/dht11_client/dht11_client.ino
ノード用 (LoRa Mini)
- LoRaToThingSpeak/dht11_server/dht11_server.ino
ゲートウェイ用((O)LG01-P-JP)

- ThingSpeak

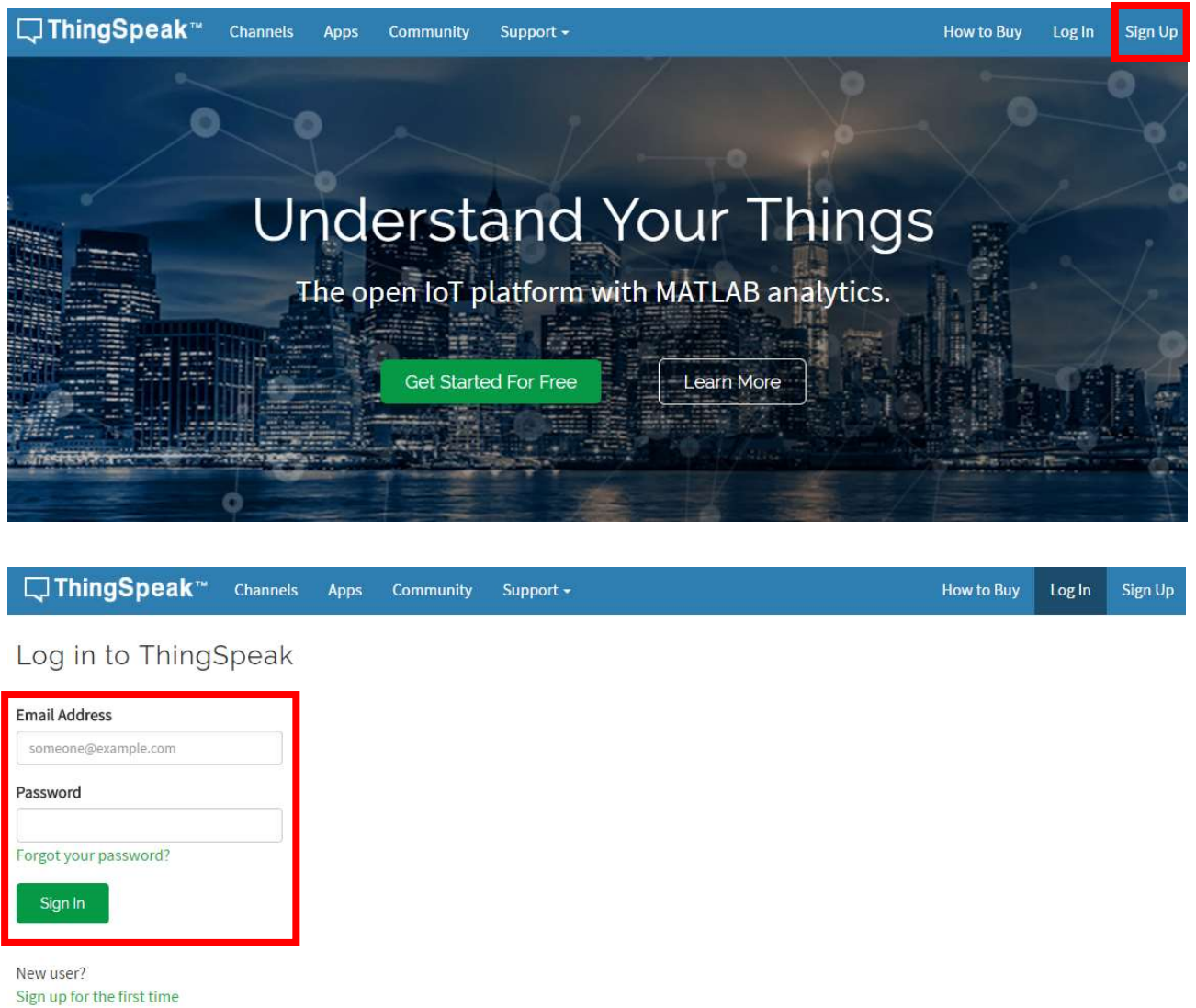
<https://thingspeak.com/>

4. IoT サーバにアップロードするまでの操作手順

4.1. ThingSpeak の設定

下記 URL から ThingSpeak にアクセスし、アカウントを作成します。

<https://thingspeak.com/>



The screenshot shows the ThingSpeak website. The top navigation bar includes links for Channels, Apps, Community, Support, How to Buy, Log In, and Sign Up. The main banner features the text "Understand Your Things" and "The open IoT platform with MATLAB analytics." Below this are buttons for "Get Started For Free" and "Learn More".

Below the banner, the "Log in to ThingSpeak" section is visible. It contains a form with the following fields:

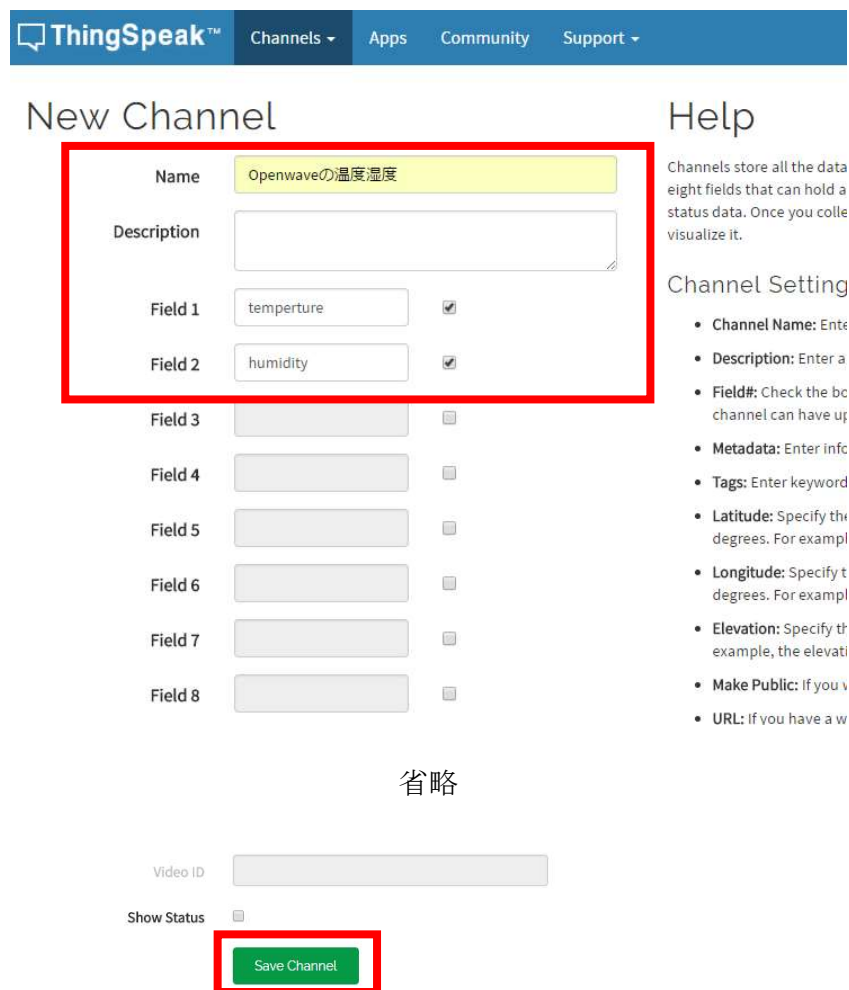
- Email Address:
- Password:
- Forgot your password? (link)
- Sign In (button)

Below the form, there is a link for "New user? Sign up for the first time".

図 3. ThingSpeak のアカウント作成

アカウントを作成後、ログインします。

My Channels の画面からチャンネルを作成します。

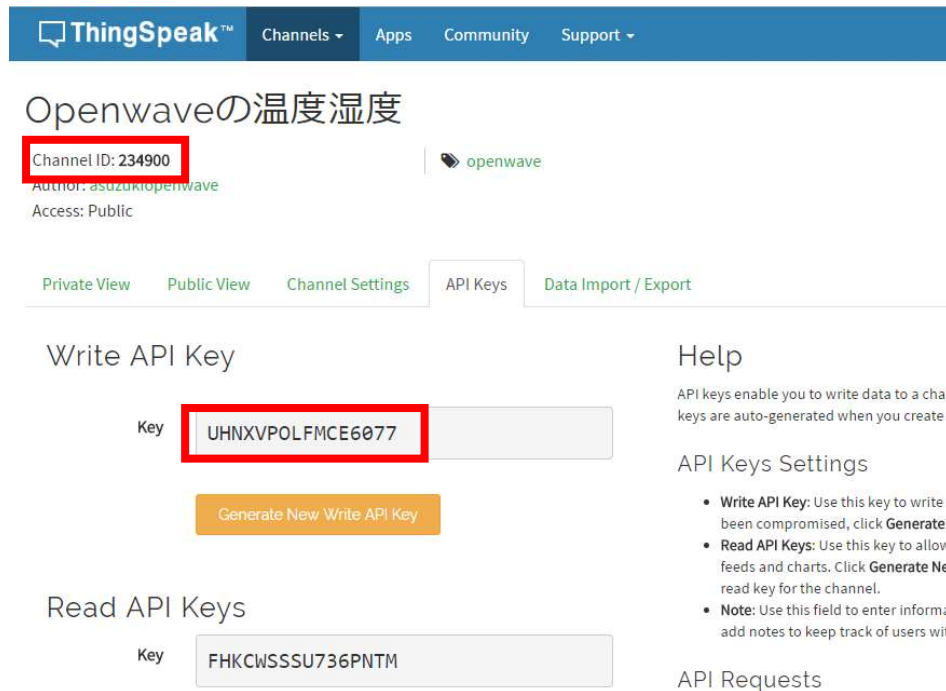


The screenshot shows the 'New Channel' form in the ThingSpeak interface. The form is titled 'New Channel' and includes a 'Help' section on the right. The form fields are highlighted with a red box. The fields are: 'Name' (containing 'Openwaveの温度湿度'), 'Description' (empty), 'Field 1' (containing 'temperture' with a checked checkbox), 'Field 2' (containing 'humidity' with a checked checkbox), 'Field 3' through 'Field 8' (all empty with unchecked checkboxes). Below the form, there is a 'Video ID' field, a 'Show Status' checkbox, and a green 'Save Channel' button highlighted with a red box.

省略

図 4.チャンネル作成

作成したチャンネルの画面を開き、チャンネルの Channel ID と Write API Key を控えます。



The screenshot shows the ThingSpeak interface for a channel named 'Openwaveの温度湿度'. The Channel ID is 234900, which is highlighted with a red box. Below the channel name, there are tabs for Private View, Public View, Channel Settings, API Keys, and Data Import / Export. The API Keys tab is selected, showing the Write API Key UHNXVPOLFMCE6077 (also highlighted with a red box) and a button to generate a new key. Below this, the Read API Key is shown as FHKCWSSSU736PNTM. On the right side, there is a Help section and API Keys Settings with instructions on how to use the keys.

ThingSpeak™ Channels Apps Community Support

Openwaveの温度湿度

Channel ID: 234900

Author: asuzukioopenwave

Access: Public

Private View Public View Channel Settings API Keys Data Import / Export

Write API Key

Key UHNXVPOLFMCE6077

Generate New Write API Key

Read API Keys

Key FHKCWSSSU736PNTM

Help

API keys enable you to write data to a channel. Keys are auto-generated when you create a channel.

API Keys Settings

- **Write API Key:** Use this key to write data to the channel. If the key has been compromised, click **Generate New Write API Key**.
- **Read API Keys:** Use this key to allow feeds and charts. Click **Generate New Read API Key** for the channel.
- **Note:** Use this field to enter informal notes to keep track of users with access to the channel.

API Requests

図 5.チャンネル情報の取得

4.2. LoRa プログラミングの前に

次のライブラリを Arduino IDE にインクルードしておく必要があります。

- 「ThingSpeak」 (ThingSpeak ライブラリ)
- 「DHT sensor library」 (DHT ライブラリ)
- 「Adafruit Unified Sensor required」 (DHT ライブラリが使用するライブラリ)

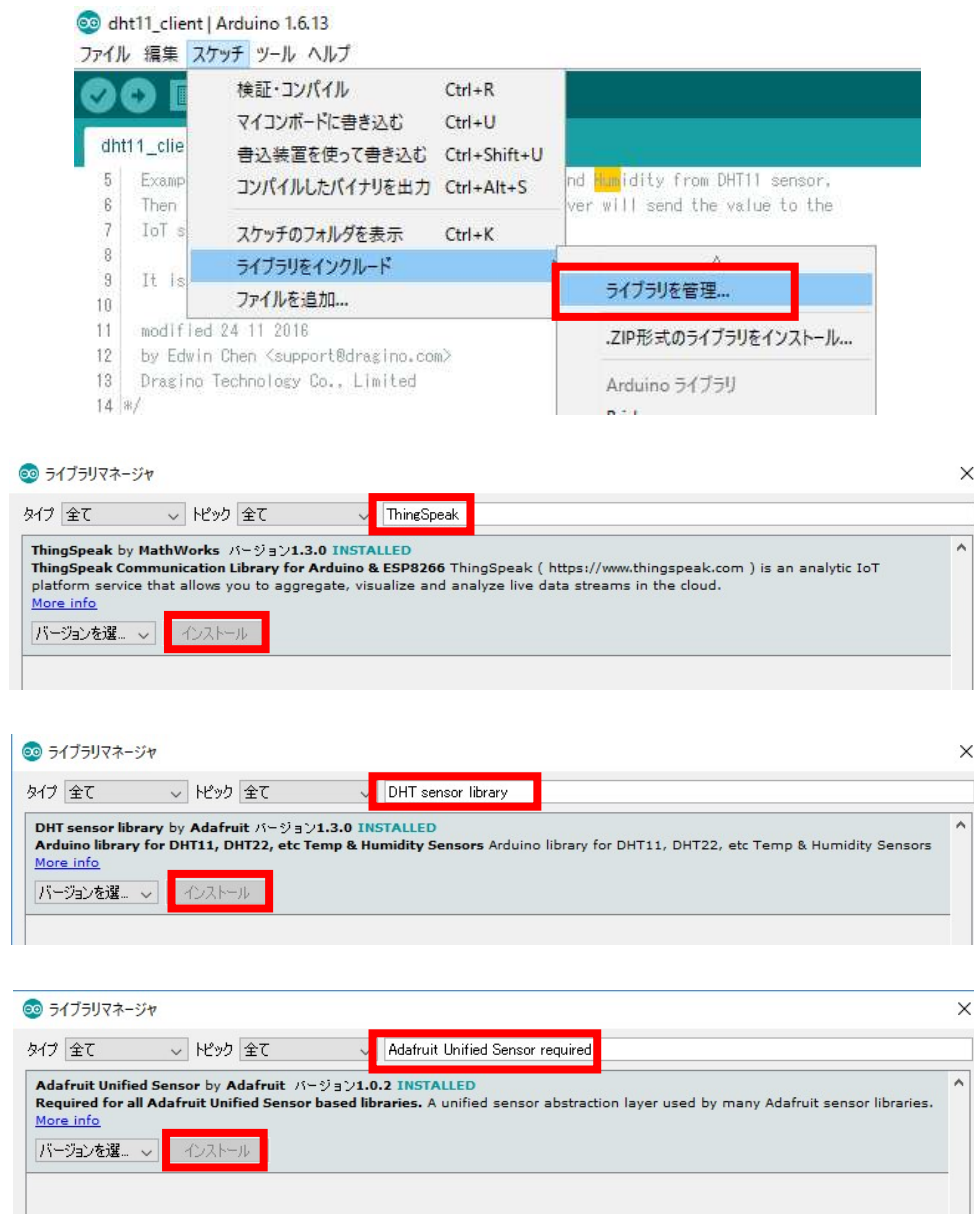


図 6.ライブラリマネージャによるインクルード

RadioHead ライブラリをインターネットから取得して、インクルードしておく必要があります。

<https://github.com/dragino/RadioHead/archive/master.zip>

こちらのファイルを解凍し、Arudino IDE の libraries フォルダに配置します。

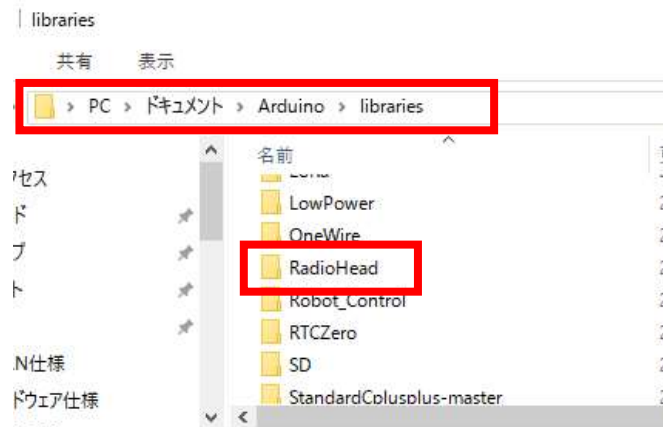


図 7. libraries フォルダにライブラリの配置

4.3. LoRa ゲートウェイのプログラミング

「使用プログラム」の dht11_server.ino を、Arduino IDE で開きます。（事前に「3 使用するもの」の Arduino IDE の設定が必要となります）

チャンネル ID と書き込みキー（Write API Key）を「図 5. チャンネル情報の取得」で取得したものに書き換えます。

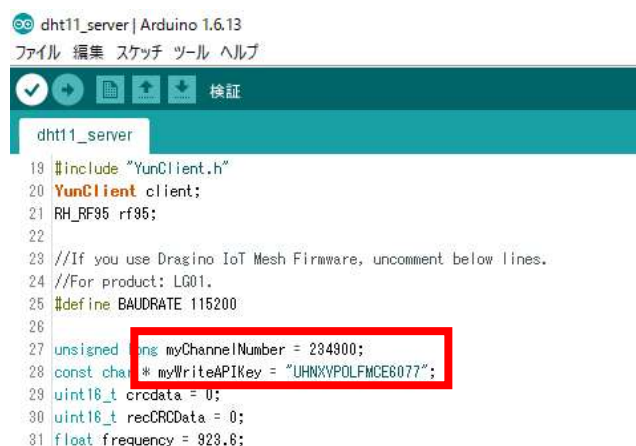


図 8. ゲートウェイプログラムの修正箇所

設定が次のようになっていることを確認し、(O)LG01-P-JP に書き込みます。

ボード : Dragino Yun + UNO or LG01/OLG01

シリアルポート : (O)LG01-P-JP の IP アドレスであること



図 9.ゲートウェイプログラムの書き込み

注意) ゲートウェイ側のプログラムで、ThingSpeak のライブラリを使いますが、ThingSpeak のライブラリ内部で、Console という名の処理を使っているため、プログラム上でログを出力するための Console を記述することはできません。(Console.print()を記述するとデータがアップされません)

4.4. LoRa ノードのプログラミング

LoRa Mini と PC を USB ケーブルで接続し、「使用プログラム」の dht11_client.ino を、Arduino IDE で開きます。

設定が次のようになっていることを確認し、LoRa Mini に書き込みます。

ボード : Arduino/Genuino Uno

シリアルポート : COMx (x は数字)



図 10. ノードの書き込み

4.5. LoRa Mini と温度湿度センサーをつなぐ

LoRa Mini と温度湿度センサーを接続します。

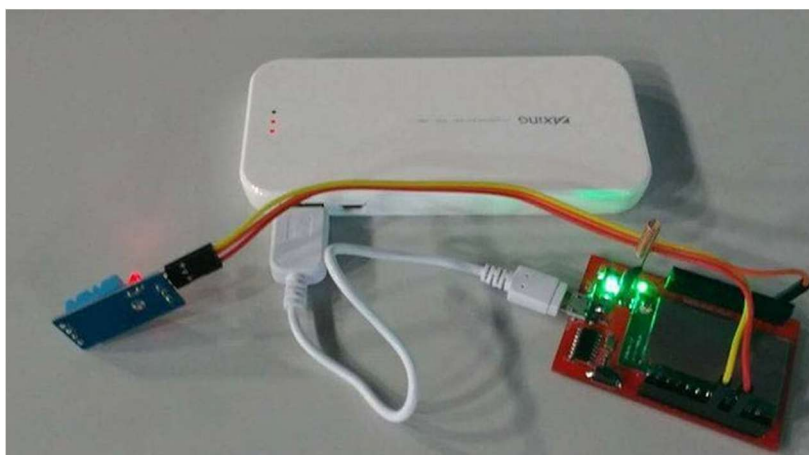


図 11.温度湿度センサーの接続

表 1.接続端子

No.	LoRa mini Dev 端子	DHT11 端子
1	3.3V	VCC
2	GND	GND
3	A0	DATA

LoRa mini Dev の仕様につきましてはこちらをご覧ください。

<http://www.dragino.com/products/lora/item/126-lora-mini-dev.html>

4.6. 接続確認

再度 ThingSpeak にアクセスし、作成したチャンネルにデータがプロットされることを確認します。

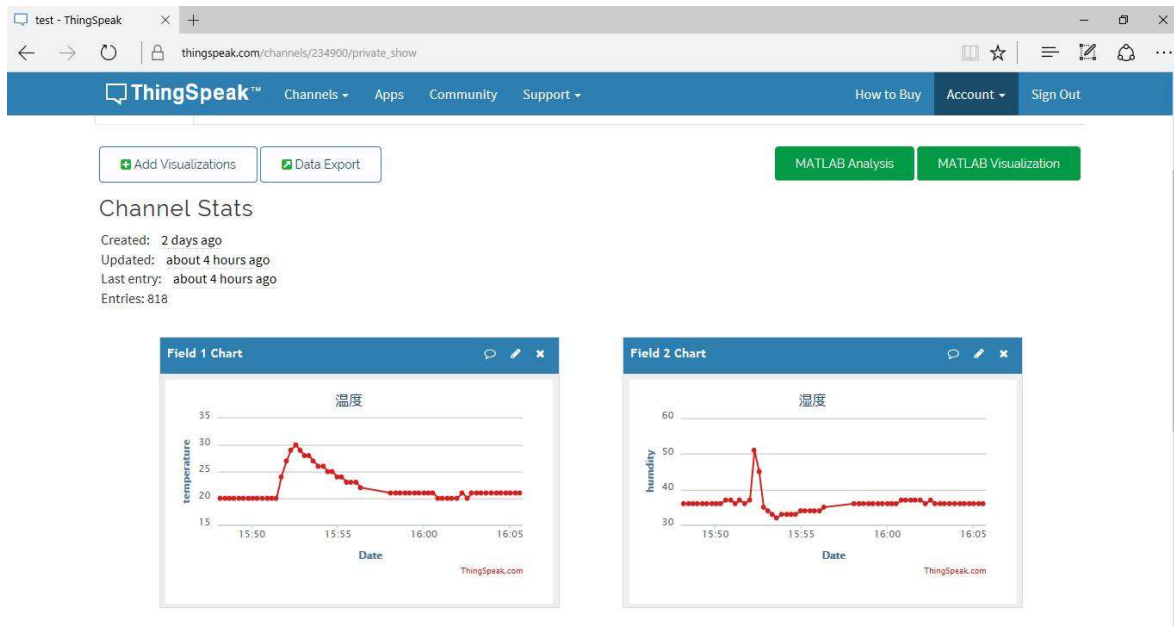


図 12.接続確認



LG01 LoRa Gateway User Manual Document Version: 1.3

Firmware Version: IoT Mesh v4.3.0

Version	Description	Date
0.1	Initiate	2016-Oct-29
1.0	Release, Add ThingSpeak Examples	2016-Dec-9
1.0	日本語翻訳	2017-Jan-9
1.1	工場時初期化方法の追記	2017-Aug-1
1.2	ダウンロード URL の変更	2017-Oct-31
1.3	英語マニュアル v1.3 ベースに修正	2017-Nov-21
1.4	USB 接続時に COM が表示される点を追記	2018-Jan-16

当マニュアルは下記 URL よりダウンロードできます。

https://www.openwave.co.jp/LG01/LG01_LoRa_Gateway_Manual_JP.pdf

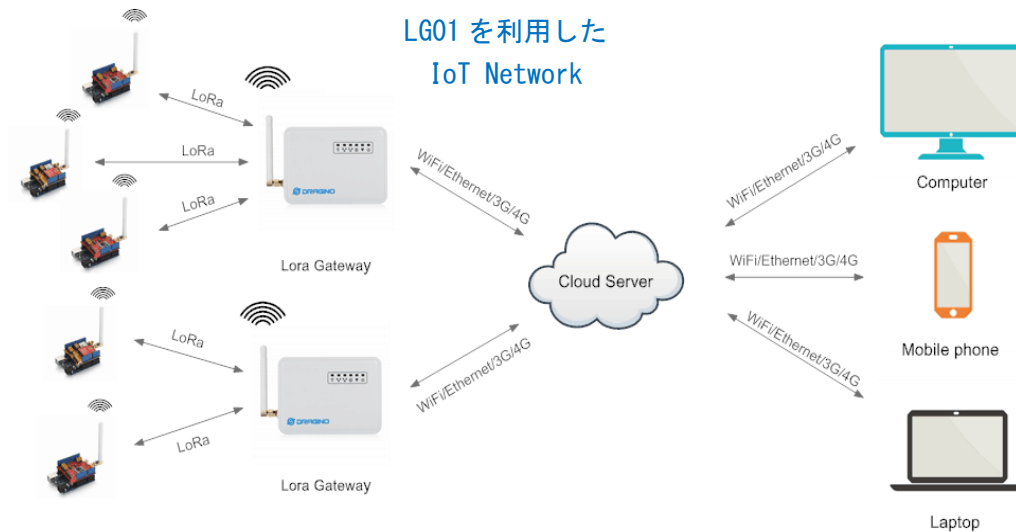
1.	イントロダクション	4
1.1	LG01 とは？	4
1.2	仕様	4
1.3	特長	6
1.4	システム構成	6
1.5	IoT アプリケーション	7
2.	クイックスタートガイド	8
2.1	アクセスと LG01 設定	8
2.2	マイクロコントローラー・プログラム	9
2.2.1	ダウンロードと Arduino IDE 設定	9
2.2.2	スケッチを MCU にアップロード	11
2.3	シンプル LoRa 無線	12
2.3.1	LoRa ライブラリをインストール	13
2.3.2	LoRa クライアントスケッチをアップロード	14
2.3.3	LoRa サーバースケッチをアップロード	15
2.3.4	実行結果について	16
3.	典型的なネットワーク設定	18
3.1	概要	18
3.2	一般的 WiFi アクセスポイントネットワーク	19
3.3	WAN ポートインターネットモード	20
3.4	WiFi クライアントモード	20
3.5	Mesh WiFi ネットワーク	21
3.5.1	Mesh ゲートウェイ設定	21
3.5.2	Mesh クライアント設定	23
3.6	USB 3G/4G モデム設定	26
3.7	USB 3G/4G Ethernet ドングル	27
4.	Linux システム	29
4.1	Linux コンソール用 SSH アクセス	29
4.2	編集とファイル転送	30
4.3	ファイルシステム	30
4.4	パッケージメインテナンスシステム	30
5.	ブリッジライブラリ	32
5.1	コンソールの使用	33
6.	アドバンスマネージメント	34
6.1	ネットワークリセット又は工場規定値にリセット	34
7.	アップグレードファームウェア	35
7.1	ウェブ UI を経由してアップグレード	35
7.2	Linux コンソールを經由してアップグレード	35
8.	マイクロコントローラスケッチのアップグレード	37

8.1	Arduino IDE によるアップグレード	37
8.2	Web UI によるアップグレード	37
8.3	MCU の自動アップデート	38
9.	アドバンス事例	40
9.1	アップストリーム: LoRa ノードからデータ取得し、IoT サーバーへ送信 ..	40
9.1.1	ハードウェア設定	40
9.1.2	IoT サーバーアカウントの設定	40
9.1.3	スケッチをアップロード	41
9.1.4	検証結果	42
9.2	ダウンストリーム: IoT サーバーからデータ取得し、LoRa ノードに送信 ..	42
9.2.1	ハードウェア設定	42
9.2.2	talkback コマンド設定	43
9.2.3	スケッチをアップロード Upload sketches	43
9.2.4	検証結果	44
9.3	他のサンプル	44
10.	FAQ	45
10.1	なぜ、433/868/915/920 バージョンがあるのですか?	45
10.2	LG01 の LoRa バージョンの周波数は?	45
10.3	どのような種類の LoRa デバイスがゲートウェイでサポートできますか? ..	45
10.4	LG01 はいくつのノードをサポートできますか?	45
10.5	LG01 がサポートできるサーバーの種類は?	46
10.6	LG01 のために私自身のファームウェアを作るとはできますか? どこの ソースコードを見つけることができますか?	46
10.7	このデバイスのサンプルをもっと入手するには?	46
10.8	一般的な LoRa の質問に関するその他のよくある質問について	46
11.	トラブルシューティング:	47
11.1	Dragino プロファイルをダウンロードできません	47
11.2	MCU と Linux システムとのブリッジが動作しません	48
11.3	Arduino IDE が LG01 を認識できません	48
11.4	新しいパッケージをインストールするときにカーネルエラーが発生し場合 の修正方法は?	48
11.5	ファームウェアがクラッシュした場合、どうやって LG01 をどう回復できま すか? ..	49
11.6	WiFi アクセス用に LG01 を設定して IP を失った場合にどうすればよいか ..	50
11.7	Linux パートの工場初期化をするには?	51
12.	注文情報	51
13.	梱包内容	51
14.	参照	51
15.	サポート	52

1. イントロダクション

1.1 LG01 とは？

LG01 は、オープンソースベースのシングルチャネル LoRa ゲートウェイ装置です。LoRaWAN ネットワークと WiFi, Ethernet, 3G, 4G 携帯ネットワークをつなぎます。LG01 の USB ポートを携帯ネットワークモジュールに装着することで、お客様のニーズに応じて LoRaWAN ネットワークから異なる IP ネットワークに連携することができます。



1.2 仕様

ハードウェアシステム:

Linux パート:

- 400MHz ar9331 プロセッサ
- 64MB RAM
- 16MB Flash

MCU パート:

- MCU: ATmega328P
- Flash: 32KB
- SRAM: 2KB
- EEPROM: 1KB

インターフェース:

- 電源入力: 9 ~ 24v DC
- 2 x RJ45 ポート
- USB 2.0 ホストポート x 1
- 内部 USB 2.0 ホストインターフェース x 1

WiFi 仕様:

- IEEE 802.11 b/g/n
- 周波数帯域: 2.4 ~ 2.462GHz
- tx 電力:
 - ✓ 11n tx 電力: mcs7/15: 11db mcs0 : 17db
 - ✓ 11b tx 電力: 18db
 - ✓ 11g 54M tx 電力: 12db
 - ✓ 11g 6M tx 電力: 18db
- WiFi 感度
 - ✓ 11g 54M : -71dbm
 - ✓ 11n 20M : -67dbm

LoRa 仕様:

- 周波数帯域:
 - ✓ Band 1 (HF): 862 ~ 1020 MHz
 - ✓ Band 2 (LF): 410 ~ 528 MHz
- 168 dB maximum リンクバジェット
- +20 dBm – 100 mW コンスタント RF 出力 vs
- +14 dBm 高効率 PA
- プログラム可能なビットレート 300 kbps まで
- 高感度: down to -148 dBm.
- 防弾フロントエンド: IIP3 = -12.5 dBm.
- 優れたブロッキング免疫 (blocking immunity)
- Low RX current of 10.3 mA, 200 nA register retention.
- 61 Hz の分解能で統合されたシンセサイザー
- FSK, GFSK, MSK, GMSK, LoRaTM, OOK モジュレーション形式
- クロックリカバリ用ビットシンセサイザー搭載
- プリアンブル検出
- 127 dB ダイナミックレンジ RSSI.
- 自動 RF センスと超高速 AFC による CAD
- CRC を使用する最大 256 バイトのパケットエンジン
- 内蔵温度センサと低バッテリーインジケータ

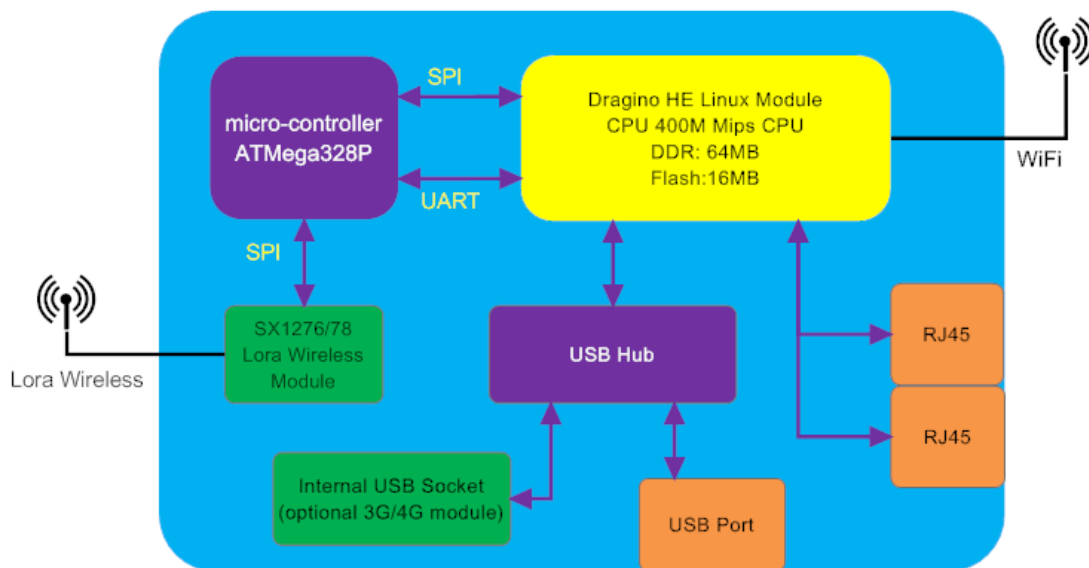
1.3 特長

- ✓ オープンソース Linux (OpenWrt) 内蔵、ユーザーがカスタマイズ、修正、ファームウェアを自由に対応化可
- ✓ 低消費電力
- ✓ Arduino IDE 1.5.4 以降互換、Arduino IDE 経由でユーザーがプログラム、デバッグ対応したスケッチを LG01 にアップロード
- ✓ ウェブ GUI, SSH 管理機能
- ✓ ネットワーク経由でソフトウェアアップグレード対応
- ✓ 自動プロビジョニング
- ✓ ウェブサーバー機能搭載
- ✓ LAN ポート, WiFi, 3G /4G 経由でネットワーク接続
- ✓ ファイルセーフデザインで堅牢システム提供

1.4 システム構成

LG01 は、Dragino オリジナル HE Linux モジュール 内で WiFi 通信機能を持ちながら、MCU (Micro Controller Unit) に ATmega328P を使い LoRa 通信機能を実現する Semtech SX1276/SC1278 通信チップを内蔵しています。

LG01 System Overview:



1.5 IoT アプリケーション



2. クイックスタートガイド

2.1 アクセスと LG01 設定

LG01 ネットワークの工場規定値は、WiFi アクセスポイントで設定されています。ユーザーは、近接している WiFi ネットワークにアクセスし設定することになります。

LG01 の初回ブート時は、アンセキュアな WiFi ネットワーク名が自動生成されます。その名称は下記となります。

dragino2-xxxxxx

ユーザーは、WiFi ネットワークに接続するためにパソコンを使うことができます。パソコン上では、IP アドレスで 10.130.1.xxx、そして LG01 の IP アドレス規定値は下記となります。

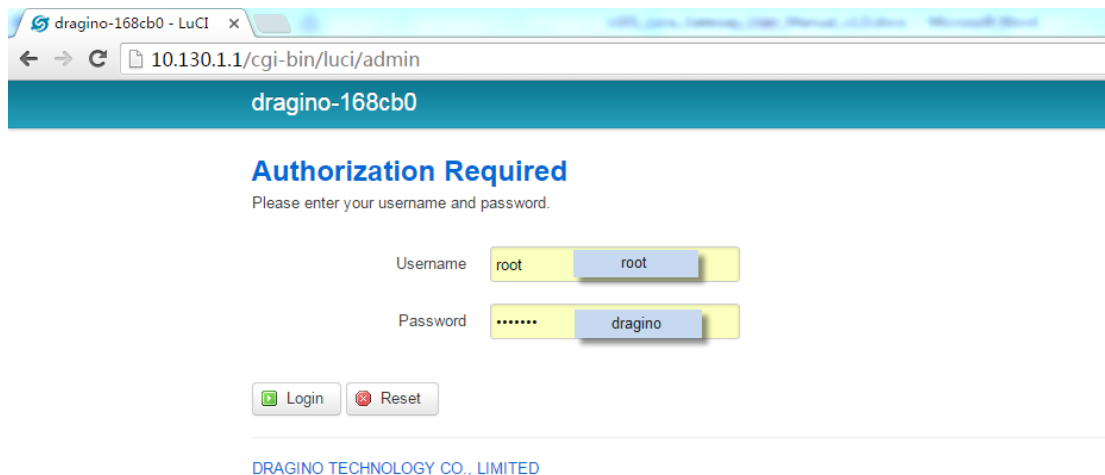
10.130.1.1

パソコン上のブラウザーを開いて、10.130.1.1 を入力します。

ユーザーには、LG01 のログイン画面が表示されます。

LG01 のウェブログインアカウントの規定値は：

Username: **root**
Password: **dragino**



2.2 マイコンコントローラー・プログラム

MCU (microcontroller) ATmega328P は、LoRa 無線と Dragino Linux モジュール HE を相互通信するために利用されます。マイコンコントローラーのプログラム言語は、C をベースとして Arduino IDE のプログラムツールを利用しています。下記は、どのようにプログラムしているかを示しています。

2.2.1 ダウンロードと Arduino IDE 設定

- 最新の Arduino IDE ソフトウェアをダウンロードします。

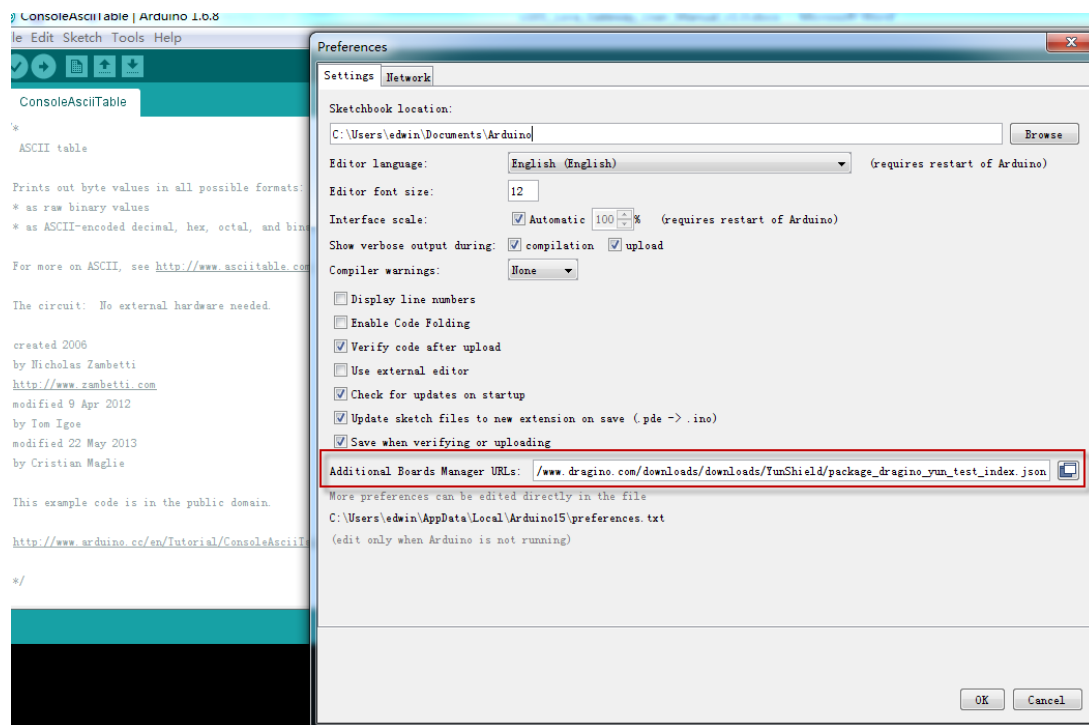
下記が Arduino 公式サイト:

<https://www.arduino.cc/en/Main/Software>

パソコンに Arduino IDE をインストールします。ファイルを開き、

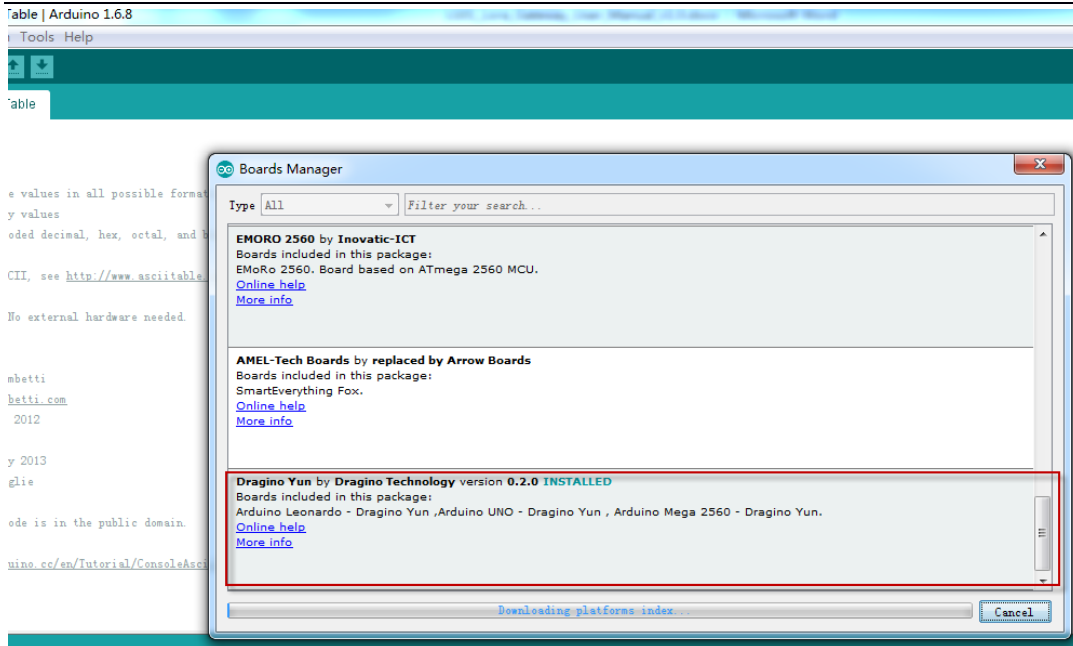
- **File → Preference**, を選択して下記 **Additional Boards Manager URLs** に下記 URL を入力します。

http://www.dragino.com/downloads/downloads/YunShield/package_dragino_yun_test_index.json



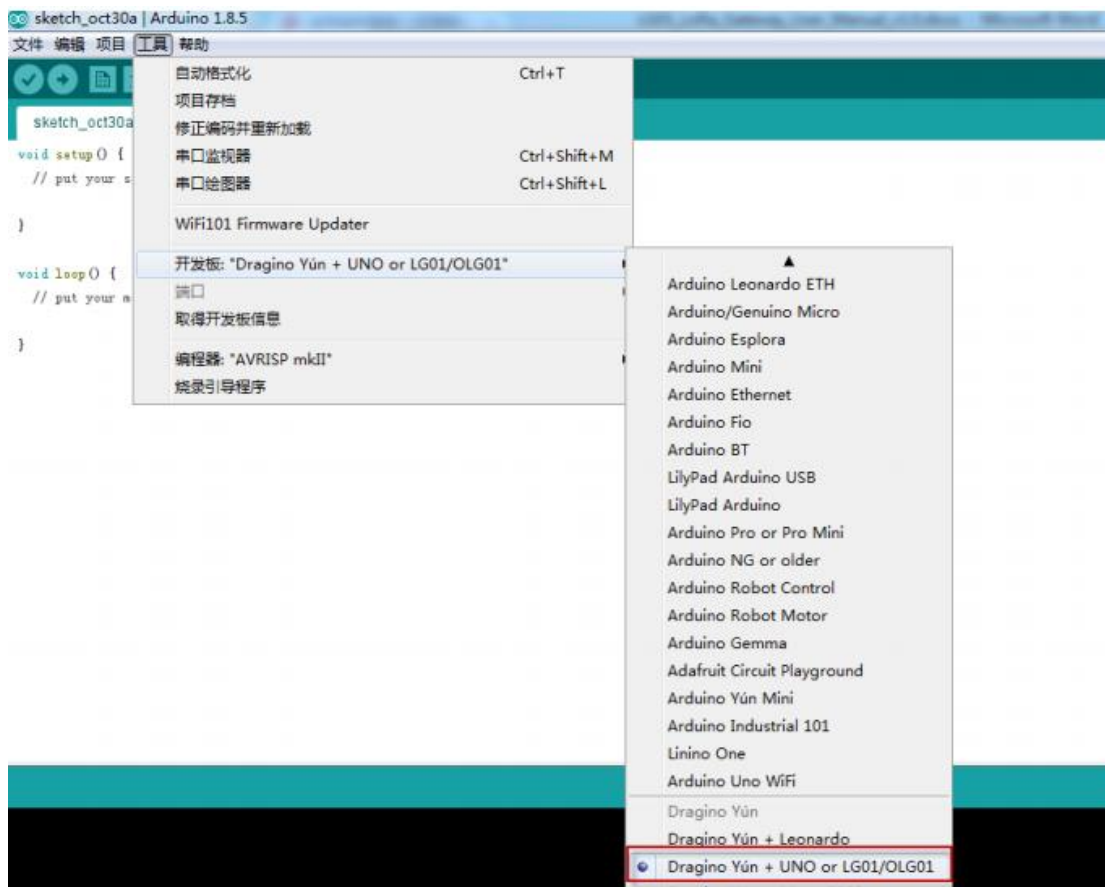
OK ボタンをクリックします。

- **tools → Boards → Boards Manager** にいき、Dragino boards info を選択してインストールします。



- Dragino board info を Arduino IDE にインストールした後、IDE 上から boards info で確認できます。LG01 の内容は下記で選択できます：

Dragino Yun-UNO or LG01/OLG01

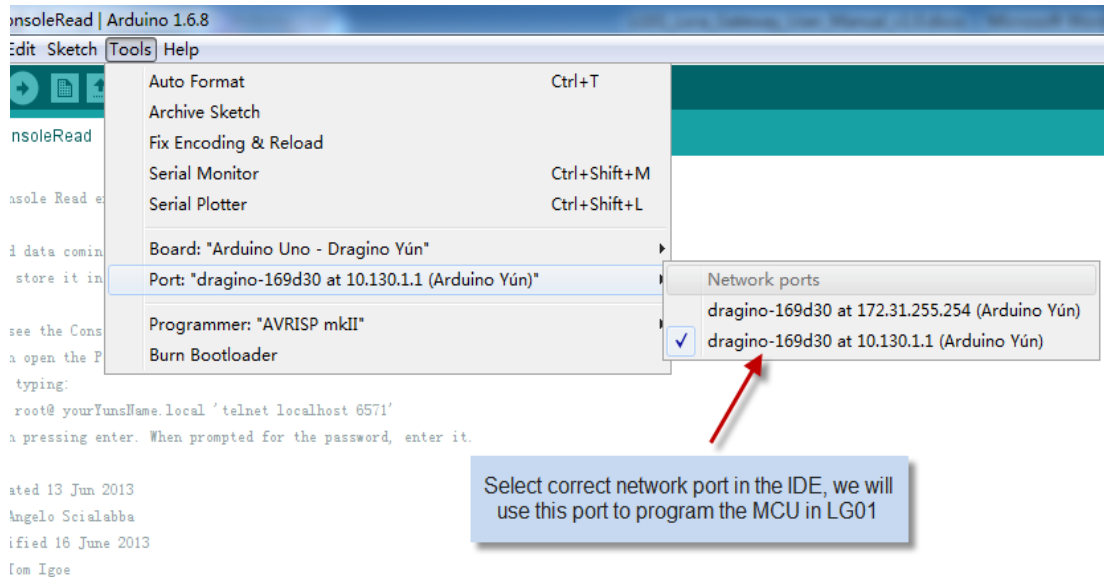


2.2.2 スケッチを MCU にアップロード

初期段階では、スケッチを MCU にアップロードする必要があります。

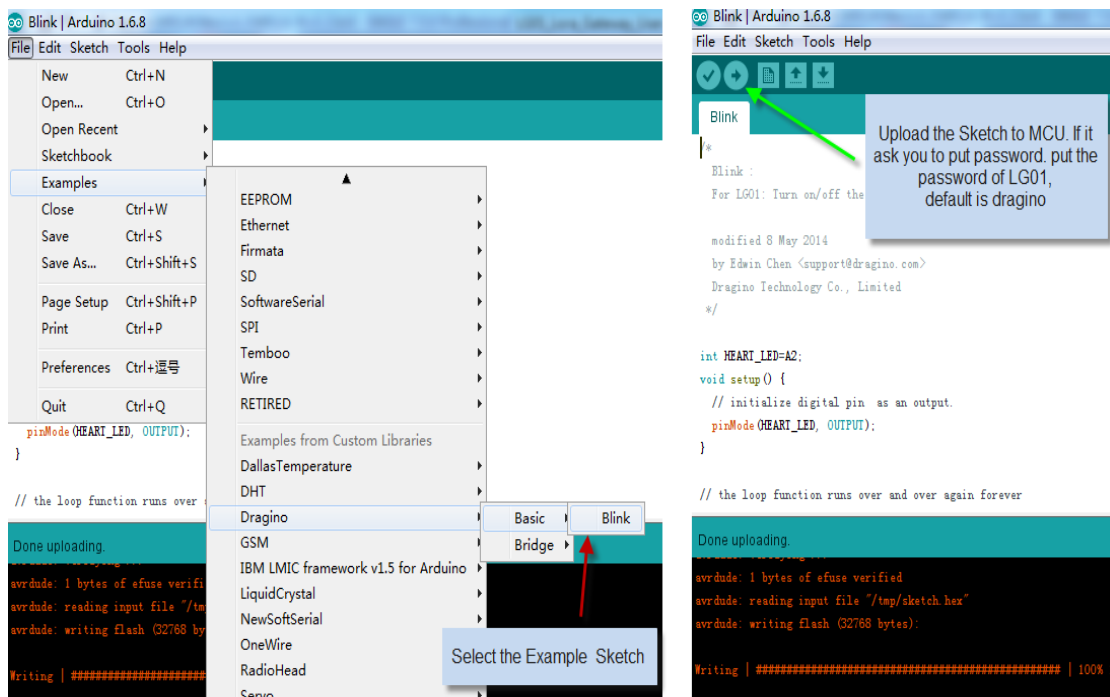
- パソコンと LG01 が同じ WiFi ネットワーク上にある必要があります。そして LG01 の SSID が既に接続してあれば、パソコンと LG01 は同じ WiFi ネットワークに存在します。

Arduino IDE の下記ポート（日本語版では「シリアルポート」）で Arduino Yun をチェックします。



そして、下記のメニューで Blink を選択します。

- **IDE → File → Examples → Dragino → Basic → Blink**
- スケッチを LG01 にアップロードするために下右画面の” → “アップロードアイコンをクリックした後に、LG01 のパスワードを入力しスケッチをアップロードします。



➤ 検証結果

ブリンクスケッチは、MCU の A2 ピンで設定できます。A2 ピンが LG01 の **HEART LED** に接続されています。もしスケッチが上手くアップロードされたら、ユーザーは定期的に **HEART LED** がオンとオフになるのを確認できます。

2.3 シンプル LoRa 無線

LoRa 無線をテストするには、少なくとも 2 つの LoRa をサポートするデバイスが必要となります。下記のような構成でテスト：

- 右側 LoRa Server：LG01
- 左側 LoRa Client：LoRa Shield + Arduino Uno



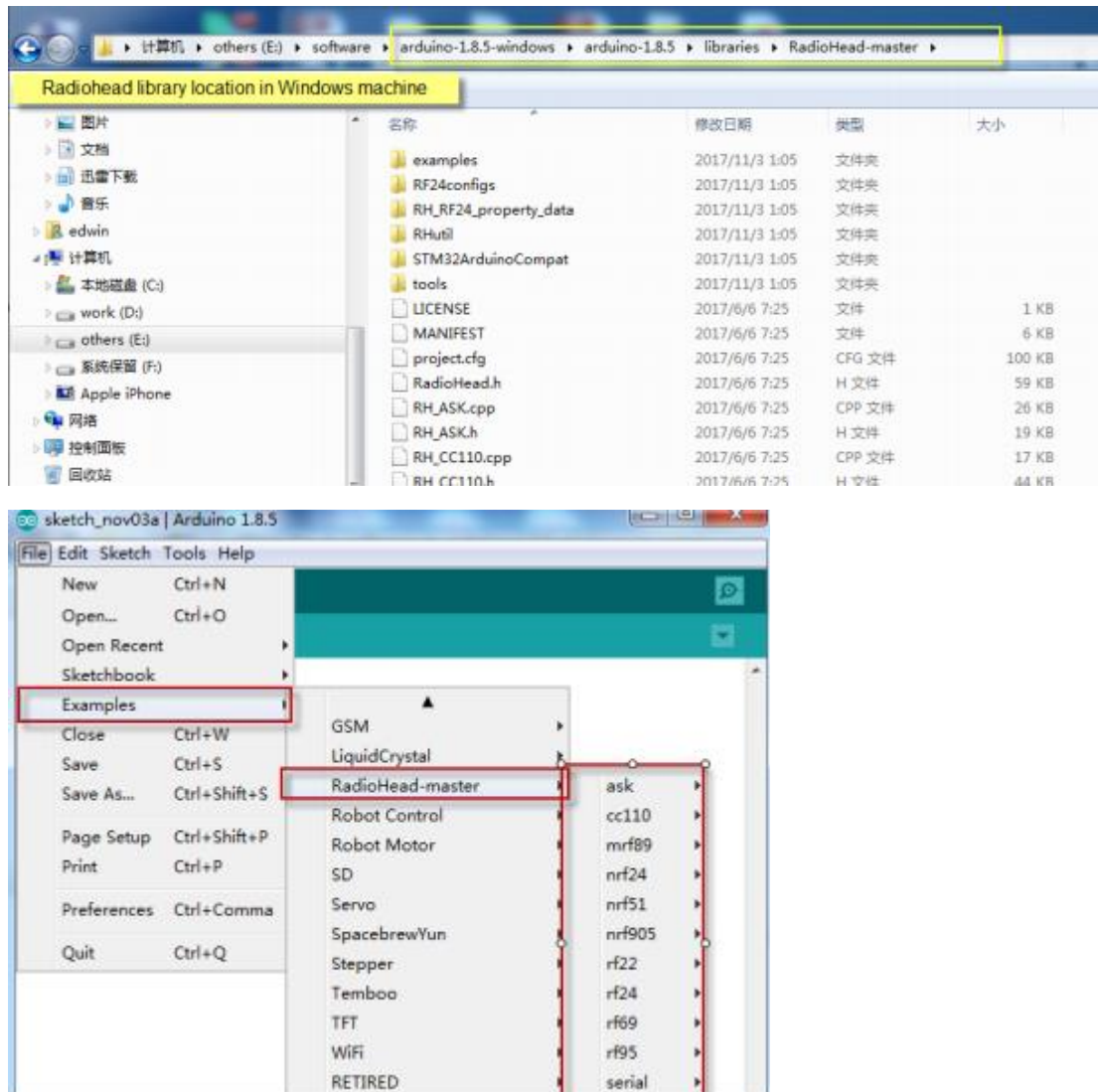
2.3.1 LoRa ライブラリをインストール

Radiohead ライブラリを使うことができます。下記 URL よりダウンロードすることができます。

<https://github.com/dragino/RadioHead/archive/master.zip>

ダウンロード後にファイルを解凍して Arduino ライブラリーフォルダーにインポートします。詳細は下記をご参照ください：

<https://www.arduino.cc/en/guide/libraries>

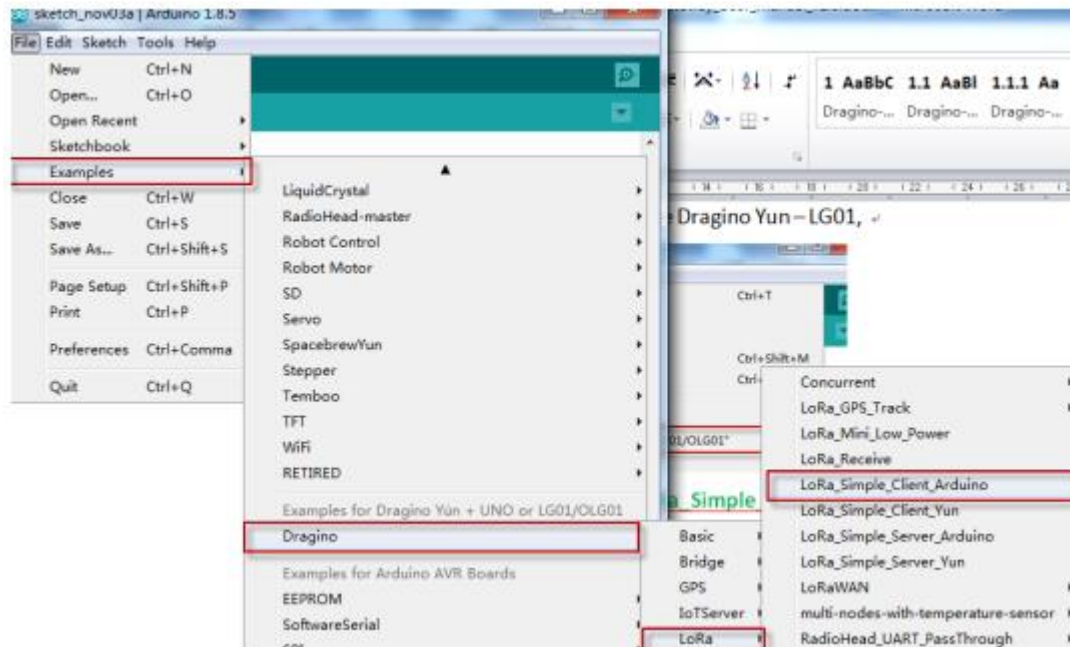


2.3.2 LoRa クライアントスケッチをアップロード

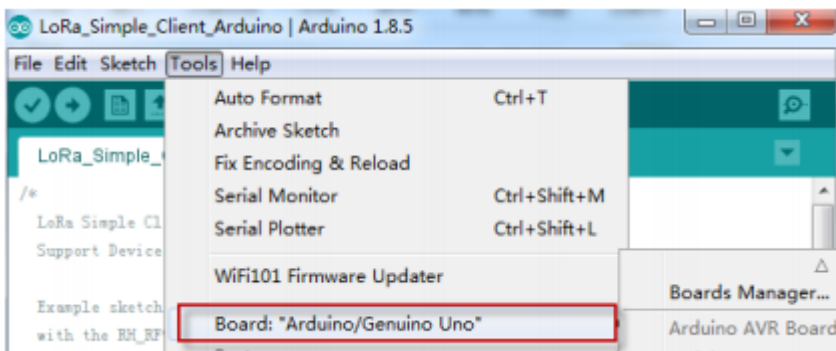
a) はじめに、Dragino Yun – LG01 を選択します。



b) 次に LoRa_Simple_Client_Arduino を選択します。



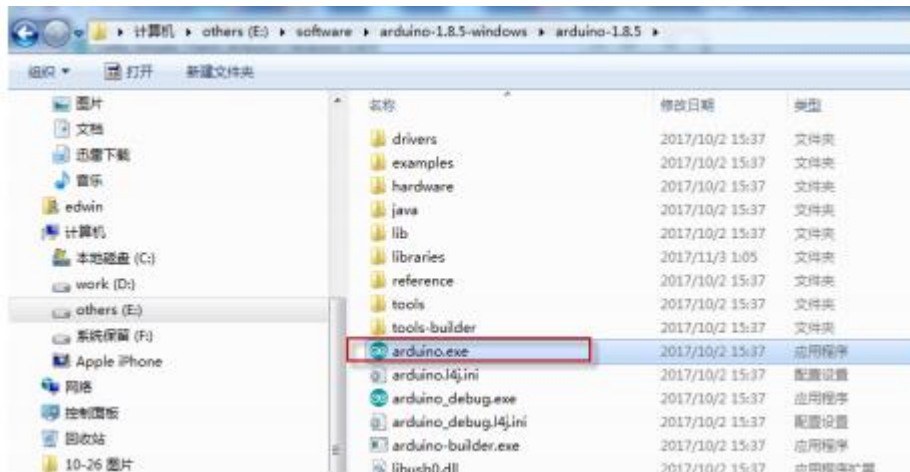
c) LoRa_Simple_Client_Arduino を開いたウィンドウで、LoRa Shield + UNO を使うために、Arduino UNO のボードを選択します。



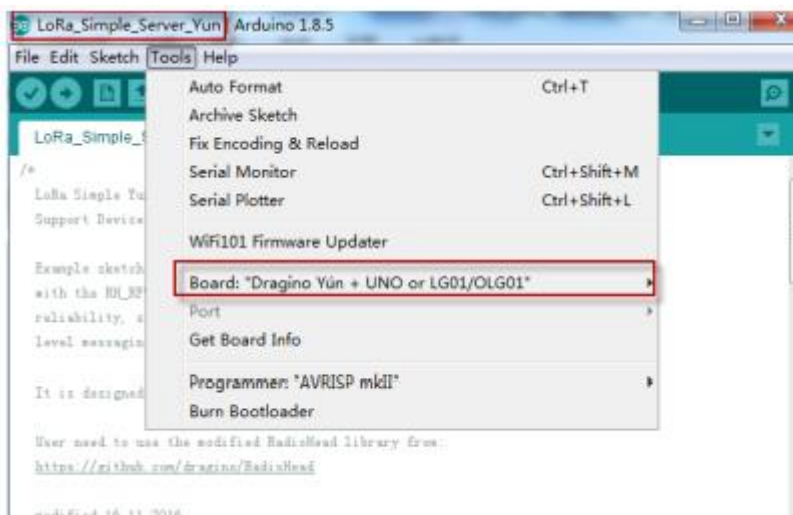
- d) LoRa Shield + UNO と PC を USB で接続すると、シリアルポートに「COM x」（x は数字）が選択できるようになります。COM x を選択しスケッチを書き込んだ後、シリアルモニタを開いて出力を確認してください。

2.3.3 LoRa サーバースケッチをアップロード

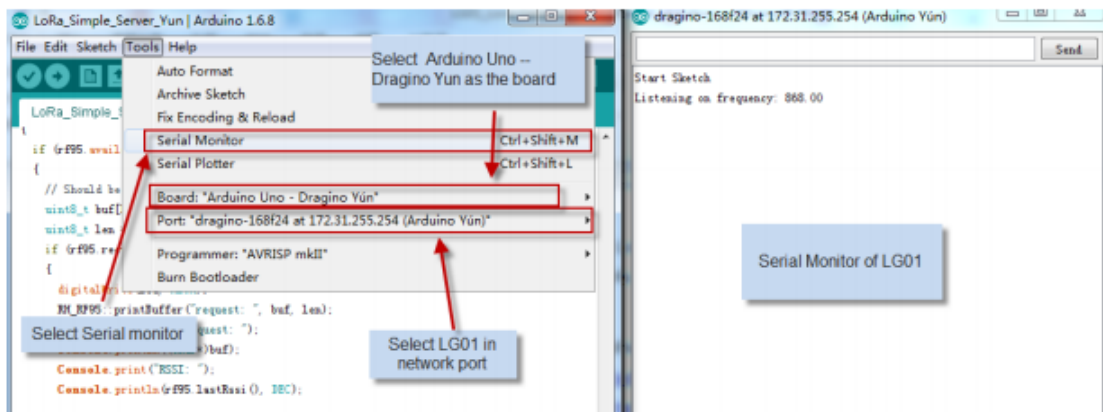
- a) LoRa Shield + UNO とは別のプロセスでサーバー用の Arduino IDE を立ち上げるため、新たに Arduino IDE を立ち上げます。



- b) 立ち上げたウィンドウで、LG01 のボードを選択し、LoRa_Simple_Server_Yun を選択します。



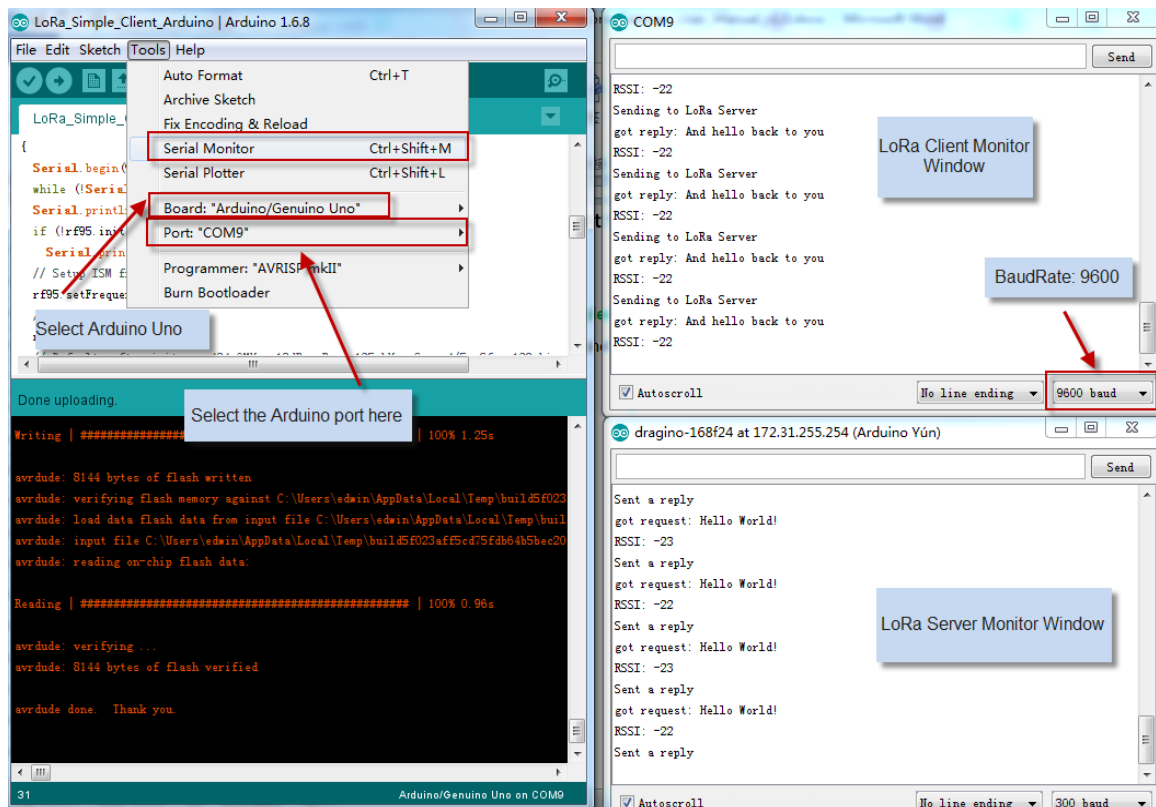
c) LG01 にスケッチを書き込み、シリアルモニタを確認します。



2.3.4 実行結果について

下記スクリーンショットが、実行した結果となります。

- 上のウィンドウは LoRa クライアントが送信を繰り返しています。
- 下のウィンドウは、LG01 が LoRa 通信から受け取った“Hello world”パケットを表示しています。LG01 はブロードキャストされた LoRa パケットに対して“and hello back to you”を返して、LoRa クライアントは受け取ったメッセージを表示しています。



注意：サンプルコードでは、LoRa クライアントに電源が一度入れられると LoRa パケットをブロードキャストします。しかし、LG01 は下記記述により、シリアルモニタを開かないと、受け取ったパケットに対して返信しません。

`while (!Console); // Wait for console port to be available`

他の LG01 を LoRa クライアント設定:

方法は同じです。下記メニューを選択します:

➤ IDE → File → Examples → Dragino → LoRa → LoRa_Simple_Client_Yun

3. 典型的なネットワーク設定

3.1 概要

LG01 は、様々な環境に対応するために柔軟なネットワークをサポートしています。この章では、LG01 で設定できる典型的なネットワークトポロジー（形態）を説明します。代表的なネットワークとして：

- ✓ WAN ポートインターネットモード
- ✓ WiFi クライアントモード
- ✓ WiFi アクセスポイントモード
- ✓ Mesh WiFi ネットワーク
- ✓ USB ダイアルアップモード
- ✓ USB イーサネットモード

ネットワークパラメータを設定する前に、ユーザーはネットワーク状況を確認するために下記 Display Net Connection で現在のネットワーク状態を確認できます。

➤ [Web](#) → [Network](#) → [Internet Access](#)

LG01 はパソコンと接続してネットワーク接続を確認することができます。そして **GLOBAL LED** でステータスを確認します。LG01 はホストに正しく接続されている場合は、**GLOBAL LED** が点滅します。

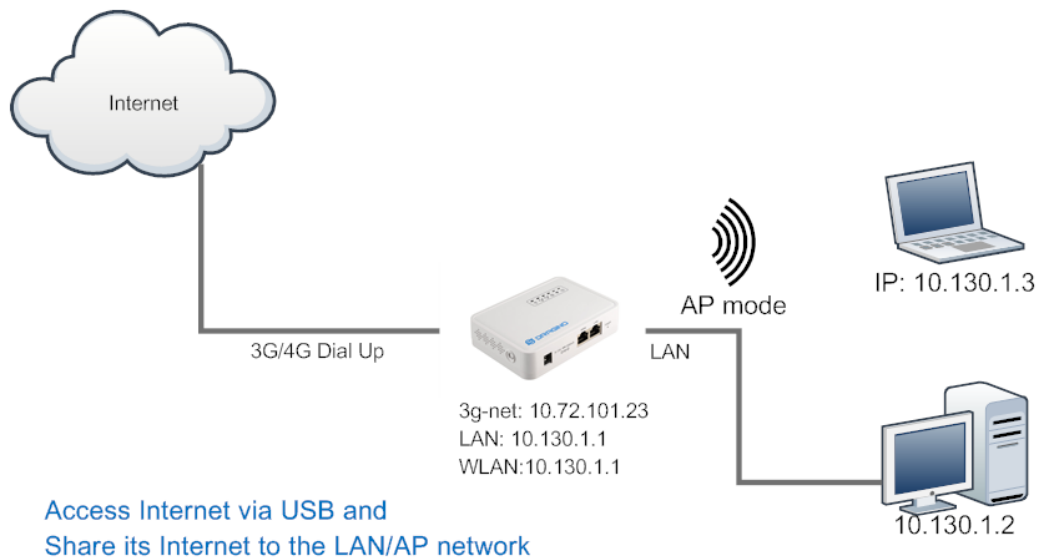
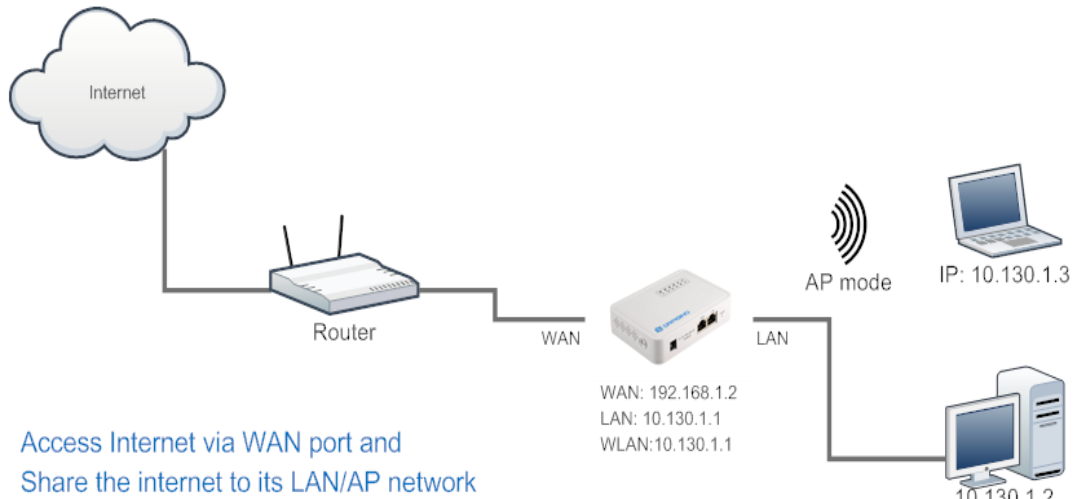


The screenshot shows the web interface for 'dragino2-3ccaef'. At the top, there are tabs: Status, System, Sensor, Network, and Logout. Below these, there are input fields for 'Gateway' and 'DNS Server' (with placeholder text 'DNS server domain or IP'). A red rectangular box highlights the 'Display Net Connection' section. Inside this box, there is a text input field containing 'www.163.com', a blue button labeled 'Check Connection to Specify Host and display via SYS LED', and a checkbox labeled 'Continuously Check Net Connection' with a question mark icon.

ネットワーク設定は、LG0 を使い典型的なネットワークの具体的な設定方法をこの章の後で詳しく述べます。

3.2 一般的WiFi アクセスポイントネットワーク

一般的なアクセスポイントモードでは、LG01 の WAN ポートか USB 接続の 3G/4G/GPRS からインターネットアクセスができます。LG01 自身が WiFi アクセスポイントとしても機能しますし、WiFi アクセスポイントネットワークを供給することもできます。LG01 は、アクセスポイントネットワークと LAN インターフェースを共用することもできます。



一般的WiFi アクセスポイントモードでウェブUI 設定

- **Network → Internet Access:**
 - ✓ WAN ポートまたは USB モデムでインターネットアクセス可能
- **Network → LAN and DHCP**
 - ✓ LAN ポートで DHCP サーバーを可能
- **Network → Access Point**

- ✓ WiFi アクセスポイントを可能
- ✓ SSID/ Encryption/ パスワードを入力

➤ **Network → Mesh Network:**

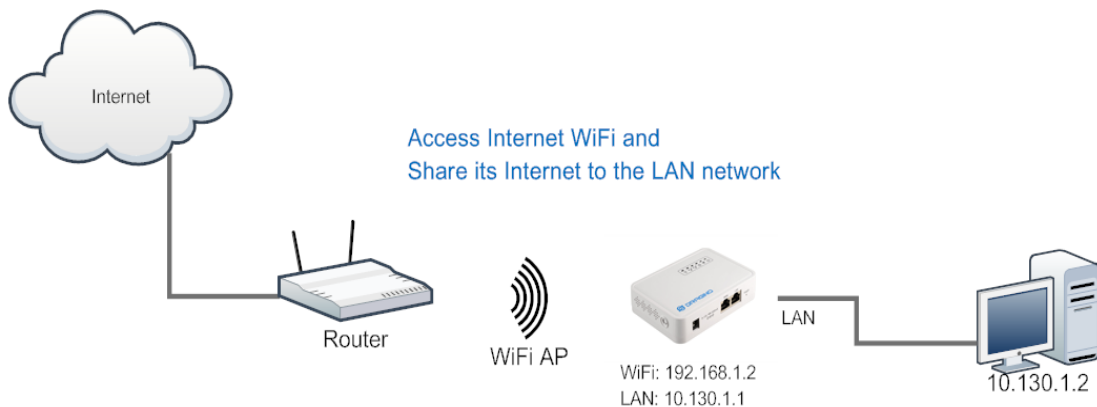
- ✓ WiFi Mesh ネットワークを可能

3.3 WAN ポートインターネットモード

LG01 は、インターネット接続用に WAN ポート を もっています。LG01 の WAN ポートを ルーターに接続する時は、ルーターから IP アドレスを割り当ててインターネットに接続 します。そして規定設定として IP01 がインターネットを LAN ポートと WiFi アクセスポ イントに共有させることができます。

3.4 WiFi クライアントモード

WiFi クライアントモードでは、IP01 は WiFi クライアントとして機能します。そして、 IP01 は WiFi 経由でルーターの DHCP 機能で IP アドレスを取得します。そして LG01 の LAN ポートを使いパソコンやデバイスとインターネット接続を共有できます。



ウェブ UI の設定

➤ **Network → Internet Access:**

- ✓ WiFi クライアント経由でインターネットアクセス
- ✓ IP アドレス取得方法: DHCP
- ✓ SSID, パスワード、暗号化 Encryption 入力

➤ **Network → LAN and DHCP**

- ✓ LAN ポート用に DHCP サーバー機能可能

➤ **Network → Access Point**

- ✓ WiFi AP 可能

➤ **Network → Mesh Network**

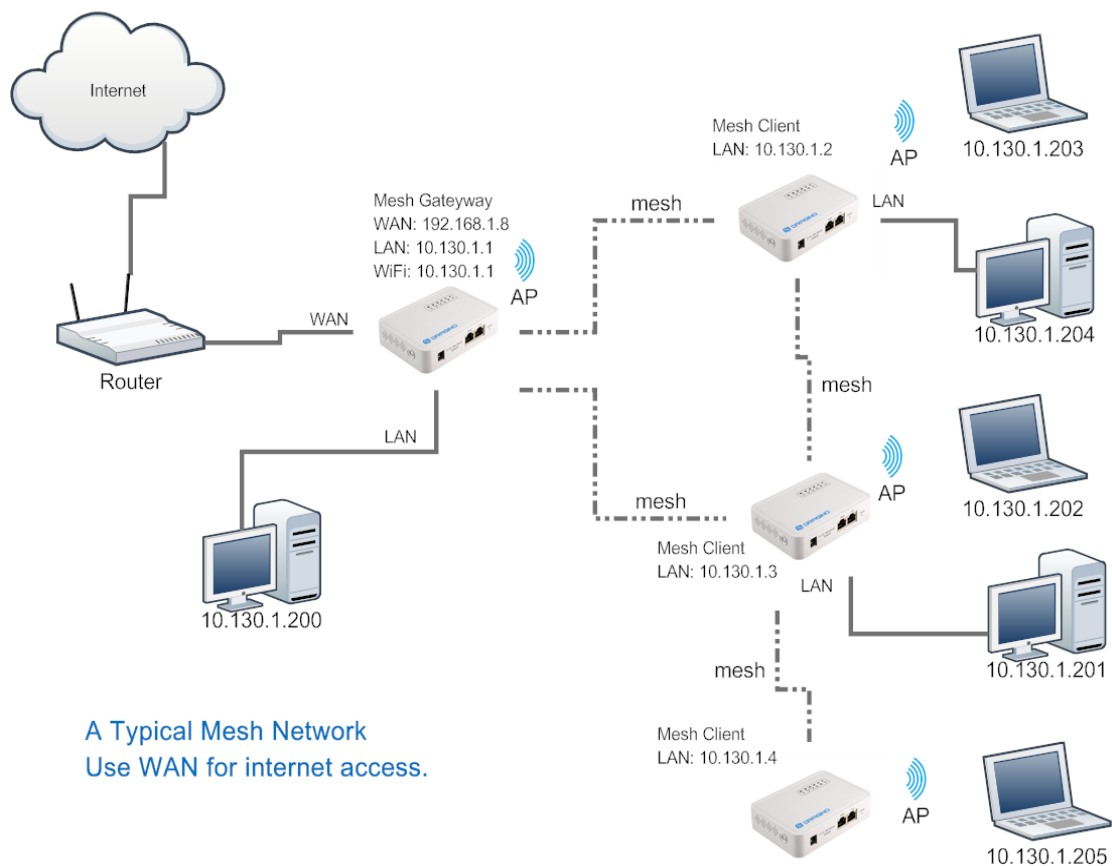
- ✓ WiFi Mesh ネットワーク可能

3.5 Mesh WiFi ネットワーク

Mesh ネットワークモードでは、ユーザーは Mesh ゲートウェイノードか Mesh クライアントノードになるべきデバイスを選択します。

Mesh ゲートウェイ：上位のルーターからインターネットアクセスするために WAN ポートまたは USB 3G/4G モデムを利用します LP01 は、インターネットを他の Mesh クライアントのために Mesh ネットワークに共有することができます。LG01 は、Mesh ゲートウェイとして Mesh ネットワーク DHCP ルーターとして機能します。

Mesh クライアント：Mesh ネットワーク経由で Mesh ゲートウェイ接続します。LG01 はさらに LAN ポートや WiFi アクセスポイントインターフェースとしてブリッジとして機能します。



A Typical Mesh Network
Use WAN for internet access.

3.5.1 Mesh ゲートウェイ設定

➤ Network → Internet Access

WAN ポートまたは USB 3G/4G モデム経由でインターネットアクセス

Small Enterprise-Campus Network

Internet Access

Access Internet Via

Way to Get IP

Display Net Connection

☒ Continuously Check Net Connection

- ✓ Mesh ゲートウェイのためにインターネット方法選択

➤ Network → LAN and DHCP

Small Enterprise-Campus Network

LAN and DHCP

Gateway Node Settings

IP Address IP Address for its LAN and AP interface.

Enable DHCP ☒ Enable DHCP Server

Authoritative ☐ Enable DHCP Authoritative

LAN Gateway Packets from LAN port and WiFi Interface (AP and Mesh) will be forward to its WAN interface

Subnet Mask

DHCP Start IP

- ✓ LAN ポートの DHCP を可能
- ✓ ゲートウェイを 255.255.255.255 に設定

➤ Network → Access Point

- ✓ WiFi アクセスポイントを可能 (not necessary), Mesh ネットワークと同じ SSID を設定

Small Enterprise-Campus Network

Access Point

Enable WiFi AP ☒ Enable WiFi AP

Station ID

Encryption

Passphrase

Channel

AP Connections

➤ Network → Mesh Network

- ✓ WiFi Mesh ネットワークを可能

- ✓ Mesh グループ入力、注記： 同じグループ内の Mesh デバイスは相互に通信することができます。Mesh グループは、容易に設定、記憶するための BSSID のための翻訳機能です。

dragino-169d30
Status ▾
Sensor ▾
System ▾
Network ▾
Logout

Small Enterprise-Campus Network

Mesh Setting

Mesh devices with the same group ID and AP wifi channel can communicate with each other

Enable Mesh ☒ ? Enable Mesh Network

Group ID

? Input a number between 1 ~ 1099511627775

Mesh Gateway

Gateway Mode

OFF ▾

3.5.2 Mesh クライアント設定

➤ [Network](#) → [Internet Access](#)

- ✓ インターネットアクセス設定を不可

dragino2-f531b1
Status ▾
System ▾
Sensor ▾
Network ▾
Logout

Small Enterprise-Campus Network

Internet Access

Access Internet Via

Disable ▾

Disable its WAN access, so packets will pass to Mesh Interface.

Display Net Connection

? Continously Check Net Connection

➤ Network → LAN と DHCP

- ✓ LAN ポートの DHCP サーバーを不可
- ✓ ゲートウェイポイントを Mesh ゲートウェイに割り当て

dragino2-f531b1 Status System Sensor Network Logout

Small Enterprise-Campus Network

LAN and DHCP

IP Address	10.130.1.2	Set a unique IP address for its LAN and WiFi interface.
Enable DHCP	☐ Enable DHCP Server	Disable DHCP server in this device.
LAN Gateway	10.130.1.1	Use the Gateway Node as Default Gateway

Enable Fallback IP ☒ Fallback IP is permanent IP in LAN port, active after reboot

➤ Network → アクセスポイント

- ✓ WiFi アクセスポイント可能（同じノード上の SSID または他の Mesh ノード上の異なる SSID）

dragino2-b170b1 Network

No password set!

There is no password set on this router. Please configure a root password to protect the web interface and Go to password configuration...

Small Enterprise-Campus Network

Access Point

Enable WiFi AP ☒ Enable WiFi AP

Station ID

Encryption

Passphrase

CAN/US Reg ☒

Channel

➤ [Network](#) → [Mesh Network](#)

- ✓ WiFi Mesh ネットワークを可能
- ✓ Mesh グループを可能、注記：同じ Mesh グループ内のデバイスは相互に通信することができます。Mesh グループは、容易に設定し記憶するために BSSID の翻訳機能です。

dragino-169d30

Status ▾

Sensor ▾

System ▾

Network ▾

Logout

Small Enterprise-Campus Network

Mesh Setting

Mesh devices with the same group ID and AP wifi channel can communicate with each other

Enable Mesh



 Enable Mesh Network

Group ID



Input a number between 1 ~ 1099511627775

Mesh Gateway

Gateway Mode

 ▾

3.6 USB 3G/4G モデム設定

LG01 の USB ポートは、3G/4G/GPRS 接続が可能になります。下記は設定のサンプルです。

注意：EC20 / UC20 モジュールを使用する場合、インターネット設定で USB モデムを選択するだけです。デバイスを再起動します。デバイスはこれらのモジュールをサポートするように自動的に構成されます。

WCDMA の設定サンプル：

dragino2-3ccaef
Network ▾

USB Modem Setting

USB Modem	Manufacturer: HUAWEI Technology, Vendor ID: 12d1, Product ID: 1436	Auto Detect USB Devices
Modem Status	inet addr: 10.72.101.23 P-t-P: 10.64.64.64 Mask: 255.255.255.255	Connection Status
Available USB Port	/dev/ttyUSB1 /dev/ttyUSB2 /dev/ttyUSB3 /dev/ttyUSB4 A modem is always detected to have several USB port for different features	
USB Modem Service	UMTS ▾	3G WCDMA
VID	12d1	Vendor ID as shown in USB info section
PID	1436	Product ID as shown in USB info section
Service APN	3gnet	Service APN. 3gnet is for China Unicom
Dial String	*99#	Dial String, Default *99#
Username		Leave blank if no provided by your provider
Password		Leave blank if no provided by your provider
PIN		Leave blank if no provided by your provider
USB Serial Port	ttyUSB1 ▾	The USB port of your dongle used for Dial Up.

3G EV-DO/CDMA2000 チャイナテレコムの設定サンプル：

dragino2-3ccaef
Status
System
Sensor
Network
Logout

USB Modem Setting

3G EV-DO dial up example:
Provider: China Telecom 3G
USB Dongle: ZTE AC582

USB Modem
Manufacturer: ZTE, Vendor ID: 19d2, Product ID: 0152

Modem Status

Available USB Port
/dev/ttyUSB0 /dev/ttyUSB1 /dev/ttyUSB2 /dev/ttyUSB3 /dev/ttyUSB4

USB Modem Service
EV-DO

Choose EV-DO

VID
19d2

Input USB dongle VID

PID
0152

Input USB dongle PID

Service APN

Dial String
#777

Dial String for Chinatelecom

Username
ctnet@mycdma.cn

User Name

Password

Password

PIN

USB Serial Port
ttyUSB0

Choose USB Serial Port for 3G

Save & Apply

3.7 USB 3G/4G Ethernet ドングル

いくつかの USB ドングルは、インターネットダイアルアップで使用できない場合があります。その代わりに、ネットワークインターフェースに内蔵されたルーター機能で対応することができます。Huawei 社 Hilink ドングルはこの典型的な例です。ユーザーがこのドングルをパソコンに装着した時、自動的にインターネットに接続し、ウェブインターフェース画面が表示されます。ドングルを Dragino 製 MS14 カードに接続する時、新しいインターフェースが表示されます。（典型的には eth2eth2 or usb0）コマンド "ifconfig -a" を実行します。ユーザーはウェブ UI を使いながら、ダイレクトにインターネット接続を設定します。

```

172.31.255.254 - SecureCRT
文件(F) 编辑(E) 查看(V) 选项(O) 传输(T) 脚本(S) 工具(L) 帮助(H)
173.236.176.38-dreamhost | 172.31.255.254
collisions:0 txqueuelen:1000
RX bytes:138038 (134.8 KiB) TX bytes:490130 (478.6 KiB)
Interrupt:5

eth1    Link encap:Ethernet  Hwaddr A8:40:41:14:31:E6
        BROADCAST MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
        Interrupt:4

eth2    Link encap:Ethernet  Hwaddr 58:2C:80:13:92:63
        inet addr:192.168.1.100  Bcast:192.168.1.255  Mask:255.255.255.0
        inet6 addr: fe80::5a2c:80ff:fe13:9263/64  Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:331 errors:0 dropped:0 overruns:0 frame:0
        TX packets:325 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:32990 (32.2 KiB)  TX bytes:26875 (26.2 KiB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128  Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1

```

new interface
from USB
modem

a new interface generate by USB Ethernet modem

dragino-1431e4 - Intern... x Editing IoT Mesh Firmwa... x +

172.31.255.254/cgi-bin/luci/stok=fd38d9c84a48a99ace8841344263d2a5/admin/network/ne

dragino-1431e4 Status System Network Logout

Small Enterprise-Campus Network

Internet Access

Access Internet Via

USB Ethernet Interface

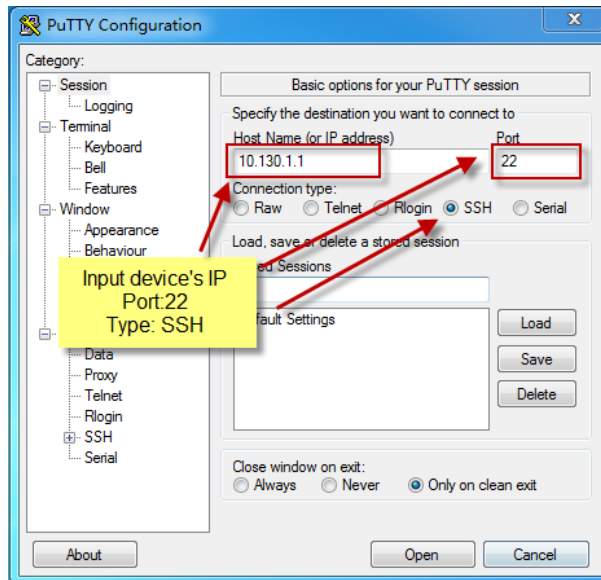
4. Linux システム

LG01 は、OpenWrt Linux システムをベースとしています。それは、オープンソースで且つ自由にユーザーが設定できます。そして Linux 設定の内部を修正もできます。

4.1 Linux コンソール用 SSH アクセス

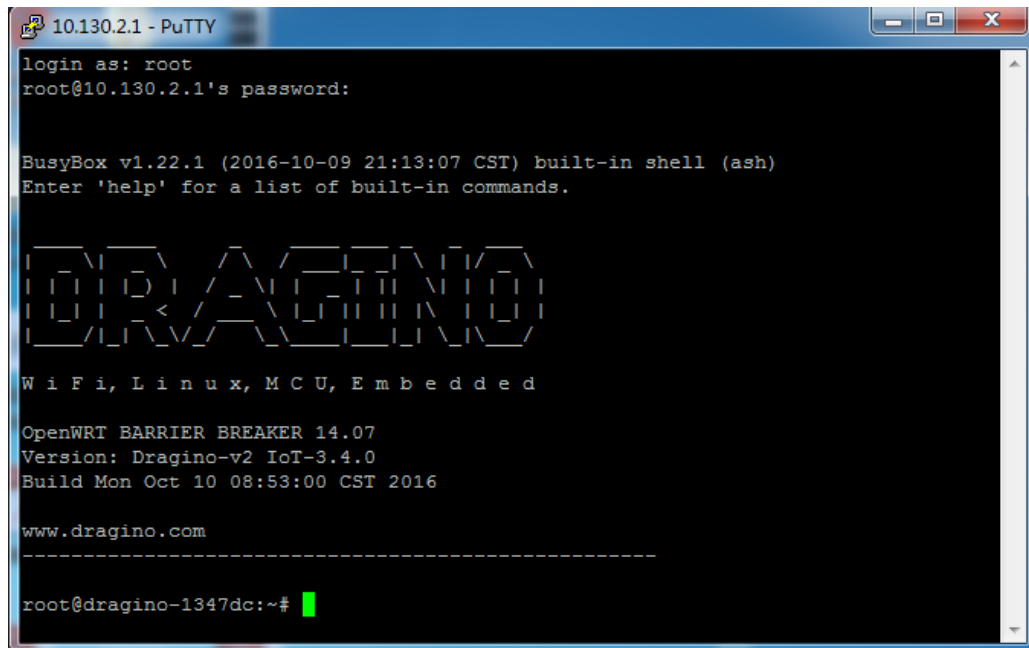
ユーザーは、SSH プロトコルを経由して Linux コンソールにアクセスすることができます。あなたのパソコンと LG01 を同じセグメントのネットワークに置きます。SSH ツール

(例: [putty](#)) を使い、アクセスします。下記がスクリーンキャプチャー画面です:



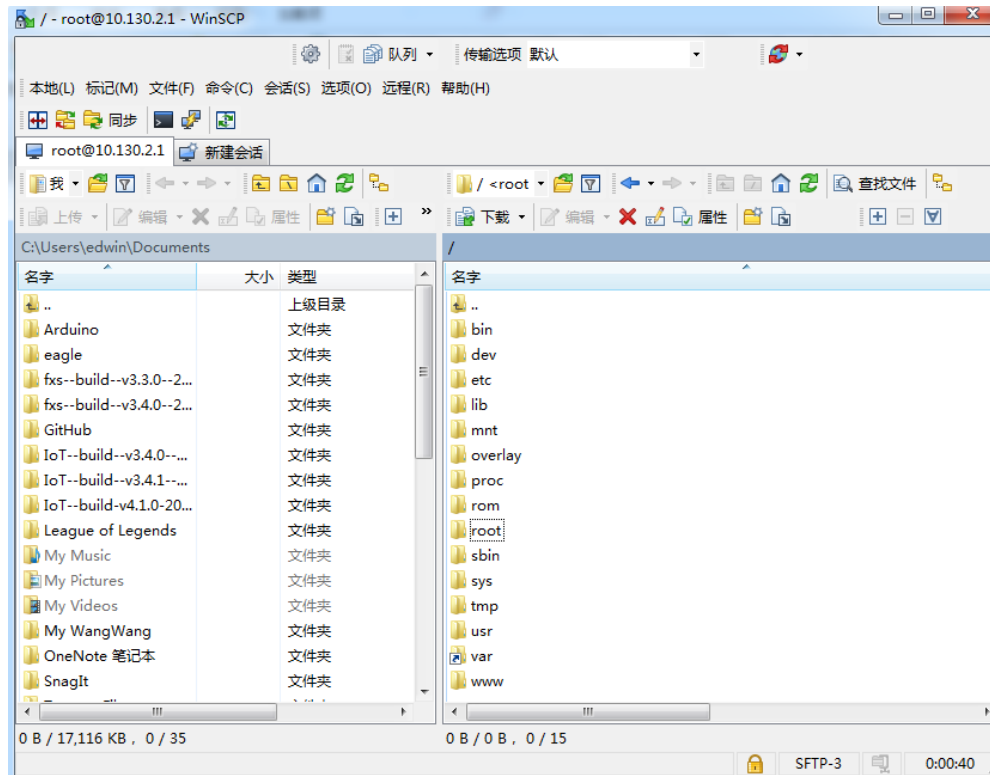
IP address: LG01 の IP アドレス
Port 番号: 22
User Name: **root**
Password: **dragino** (default)

ログインした後、Linux コンソール画面に入り、Linux コマンドが実行できます。



4.2 編集とファイル転送

LG01 は、**SCP プロトコル** をサポートしています。そして **SFTP サーバー機能** を内蔵しています。2つのプロトコルをつかい、編集しファイルを転送する方法はいくつかあります。簡単な方法の1つは、[WinSCP](#) ユーティリティを使う方法です。WinSCP を経由してデバイスにアクセスした後、ユーザーは、FTP ライクなウィンドウ画面で LG01 のファイルをドラッグ&ドロップして転送や編集ができます。下記がスクリーンキャプチャー画面です：



4.3 ファイルシステム

LG01 は、16MB flash と 64MB RAM を内蔵しています。RAM には、/var と /tmp ディレクトリがあります。デバイスをリブートした後に /tmp と /var 消去されます。他のディレクトリは、flash 内になります。そしてそれらはリブートした後も保存されています。

Linux システムは、だいたい 8MB ~10MB flash サイズです。LG01 flash 内にデータが格納されるには十分な容量がありません。ユーザーは、ストレージ用の容量を拡張するために外部の USB Flash を利用することができます。

4.4 パッケージメンテナンスシステム

LG01 は、[OPKG パッケージメンテナンスシステム](#) を利用しています。ユーザーは、3000 以上の種類のパッケージがパッケージサーバーでアプリケーション利用ができます。例えば、もし、MQTT サーバーを追加でサポートしたい場合、関連したパッケージをインストールし、MQTT をサポートするために LG01 を設定することができます。

下記は、opkg コマンドのいくつかの例です。

<https://wiki.openwrt.org/doc/techref/opkg>

Linux コンソールを実行：

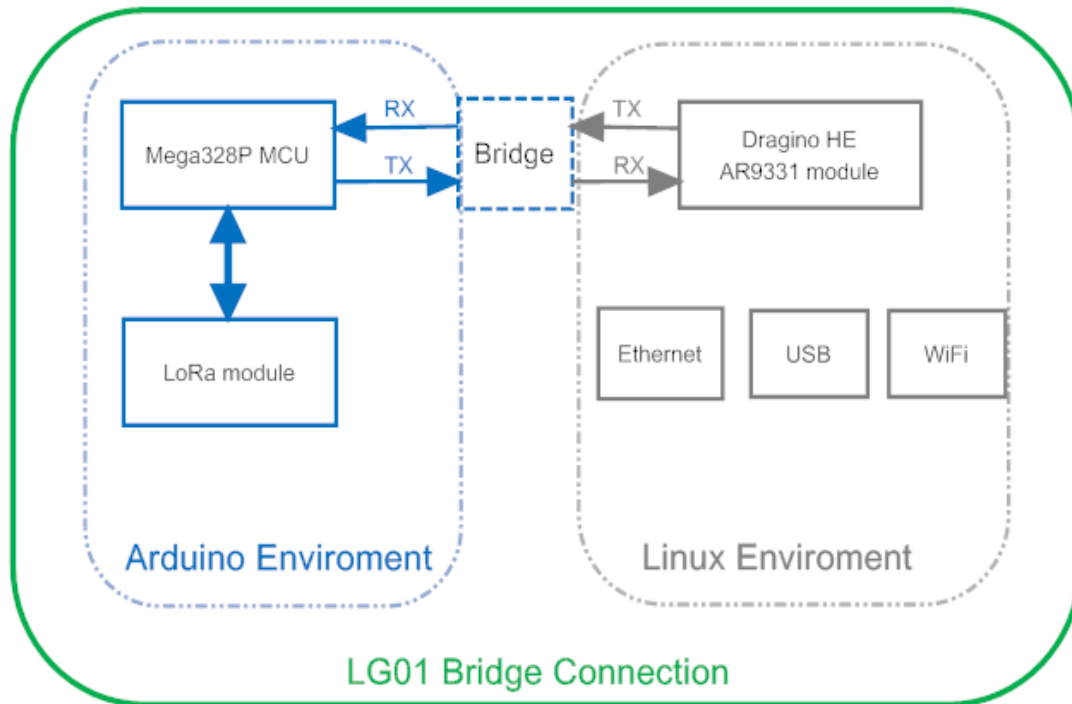
```
root@dragino-169d30:~# opkg update // 最新のパッケージリストを取得
root@dragino-169d30:~# opkg list //利用可能なパッケージを表示
root@dragino-169d30:~# opkg install mosquitto-client // MQTT クライアントをインストールし、要求したパッケージが自動インストールされます。
```

```
Installing mosquitto-client (1.3.5-1) to root...
Downloading
http://downloads.openwrt.org/barrier_breaker/14.07/ar71xx/generic/packages/packages/mosquitto-client_1.3.5-1_ar71xx.ipk.
Installing libcares (1.10.0-1) to root...
Downloading
http://downloads.openwrt.org/barrier_breaker/14.07/ar71xx/generic/packages/packages/libcares_1.10.0-1_ar71xx.ipk.
Installing libmosquitto (1.3.5-1) to root...
Downloading
http://downloads.openwrt.org/barrier_breaker/14.07/ar71xx/generic/packages/packages/libmosquitto_1.3.5-1_ar71xx.ipk.
Configuring libcares.
Configuring libmosquitto.
Configuring mosquitto-client.
```

5. ブリッジライブラリ

ブリッジライブラリは、LG01 の最も重要な機能です。ブリッジライブラリは、MCU (Mega328P) と CPU (Dragino HE AR9331) と機能連携するメカニズムです。ブリッジライブラリで、MCU はデータを CPU に送信したり、CPU からのコマンド結果を取得したり、CPU 内でコマンドを実行したりします。

ブリッジライブラリは、UART ポートを使い、MCU と CPU との間を通信します。下記は、MCU と Linux システムとのブリッジ接続を表すブロック図です。



ブリッジライブラリに関する詳細説明は下記より参照いただけます。

<https://www.arduino.cc/en/tutorial/bridge>

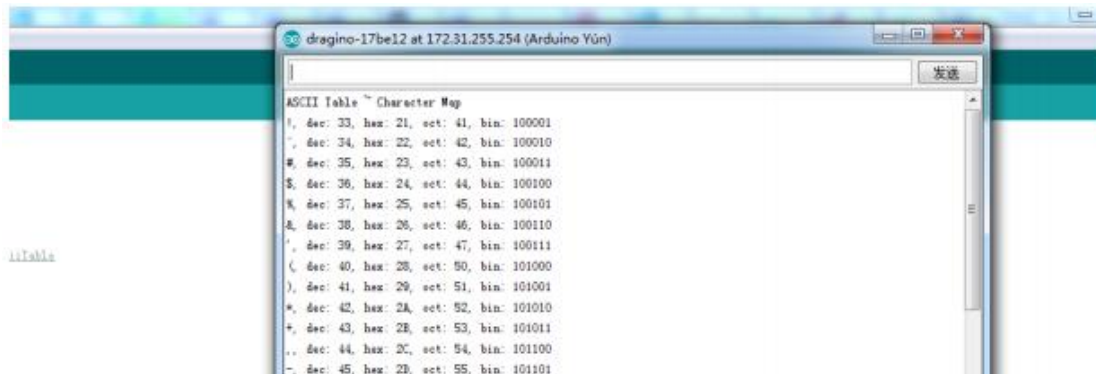
ハードウェアの違いのため Arduino サイトからブリッジ事例を参照したり、利用したりする方法はいくつかあります：

- ユーザーは、下記の実行サンプルを参照されることを推奨します。
Arduino IDE → Files → Examples → Dragino
- ブリッジクラスを利用する時は、ユーザーは、LG01 のスケッチ内の `Bridge.begin(115200)` をコールする必要があります。
- Arduino IDE からのブリッジの規定サンプルは、デバック情報を表示するために **Serial Class** を使用します。これは LG01 では機能しません。なぜなら、**Serial Class** は、MCU である ATmega328P のハードウェアのシリアルポートをコールするからです。これはブリッジライブラリと衝突するからです。もしユーザーが、デバック情報を表示する場合、`Console Class` を使用します。

5.1 コンソールの使用

Arduino のデバッグには Console クラスによる PC への情報出力が有用です。下記サンプルで、このクラスの使い方が示されています。

Arduino IDE → Files → Examples → Dragino → Bridge → ConsoleRead



シリアルモニタの代わりに LG01 に SSH 接続しての確認可能です。

root@dragino-169d30:~# telnet localhost 6571

BusyBox v1.23.2 (2017-06-24 23:34:27 CST) built-in shell (ash)

DRAGINO

WiFi, Linux, MCU, Embedded

OpenWRT Chaos Calmer 15.05
Version: Dragino-v2 IoT-4.2.2
Build wed Jul 19 15:06:00 CST 2017

www.dragino.com

root@dragino-17be12:~# telnet localhost 6571

```
ASCII Table ~ Character Map
!, dec: 33, hex: 21, oct: 41, bin: 100001
", dec: 34, hex: 22, oct: 42, bin: 100010
#, dec: 35, hex: 23, oct: 43, bin: 100011
$, dec: 36, hex: 24, oct: 44, bin: 100100
%, dec: 37, hex: 25, oct: 45, bin: 100101
&, dec: 38, hex: 26, oct: 46, bin: 100110
', dec: 39, hex: 27, oct: 47, bin: 100111
(, dec: 40, hex: 28, oct: 50, bin: 101000
), dec: 41, hex: 29, oct: 51, bin: 101001
*, dec: 42, hex: 2A, oct: 52, bin: 101010
+, dec: 43, hex: 2B, oct: 53, bin: 101011
,, dec: 44, hex: 2C, oct: 54, bin: 101100
-, dec: 45, hex: 2D, oct: 55, bin: 101101
., dec: 46, hex: 2E, oct: 56, bin: 101110
```

6. アドバンスマネージメント

6.1 ネットワークリセット又は工場規定値にリセット

LG01 では、ユーザーがデバイスをリセットする方法があります。Linux システムが実行中の時、ユーザーはデバイスをリセットするためにトグルボタンを押すことができます。ボタンを押す時間は、リセットされる部分によって変わります。

- トグルボタンを押しながら、**GLOBAL LED** が点滅します。5 秒後ボタンをリリースします。LG01 デバイスはネットワーク設定をリセットし、リブートします。
(GLOBAL/LAN/WAN/WiFi 一度点滅)、他の設定は保存されたままです。
- トグルボタンを押しながら、**GLOBAL LED** が点滅します。30 秒後ボタンをリリースします。LG01 デバイスは全ての設定が工場規定値になり、リセットします。
(GLOBAL/LAN/WAN/WiFi 一度点滅)

7. アップグレードファームウェア

新規機能やバグ修正のため LG01 Linux 側ファームウェアをアップグレードし続けることができます。最新のファームウェアは下記 URL で確認できます：

<http://www.dragino.com/downloads/index.php?dir=motherboards/ms14/Firmware/IoT/>

変更ログは下記 URL で確認できます：

http://wiki.dragino.com/index.php?title=Mesh_IoT_Firmware

dragino2-IoT-xxxxx-squashfs-sysupgrade.bin

のファイル名は、アップグレードイメージです。アップグレードするにはいくつかの方法があります：

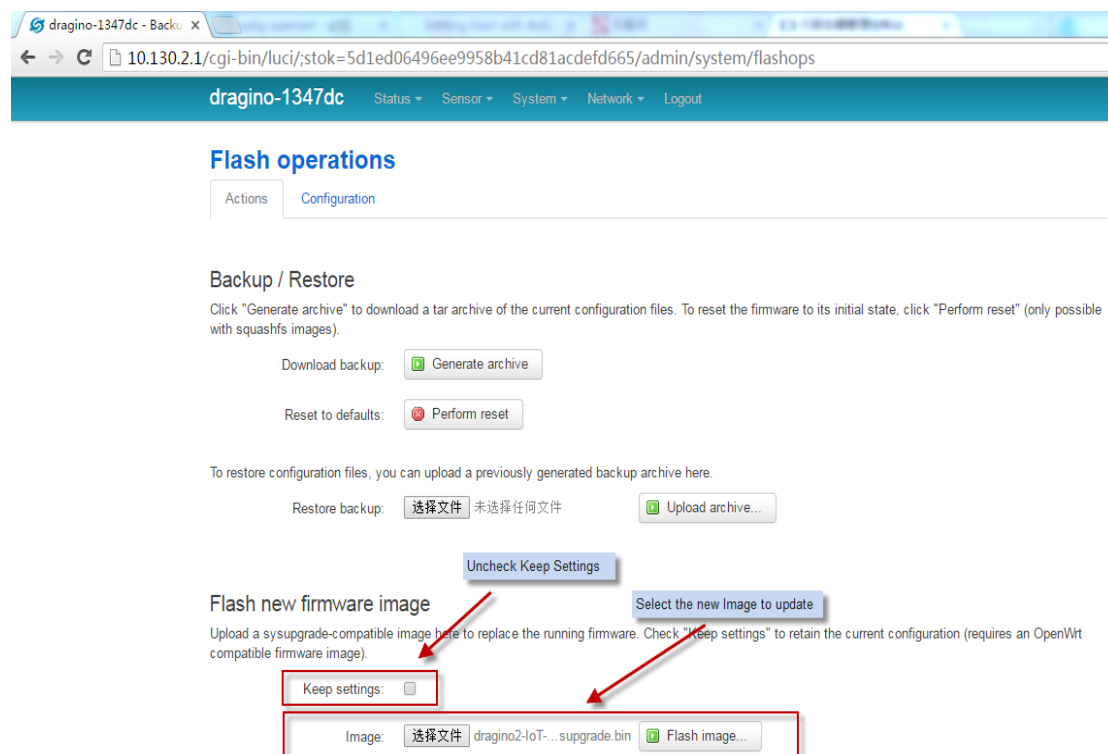
7.1 ウェブ UI を経由してアップグレード

下記のページに移動します：

➤ **Web → System → Back Up and flash firmware**

イメージファイルを選択します。そして Flash イメージを選択します。イメージは LG01 にアップロードされます。そして Process Update をクリックします。

システムは、アップグレードした後に新規ファームウェアで自動ブートされます。



7.2 Linux コンソールを經由してアップグレード

SCP ツールでファームウェアをシステムの **/var** ディレクトリに移動して実行します。

```
root@OpenWrt:~# /sbin/sysupgrade -n /var/Your_Image
```

注記！：イメージファイルを/var ディレクトリに移動するのはとても重要です。さもないと Flash サイズがオーバーフローしてしまいます。

8. マイクロコントローラスケッチのアップグレード

次の 3 つの方法があります。

8.1 Arduino IDE によるアップグレード

「2.3 シンプル LoRa 無線」でご紹介した方法です。

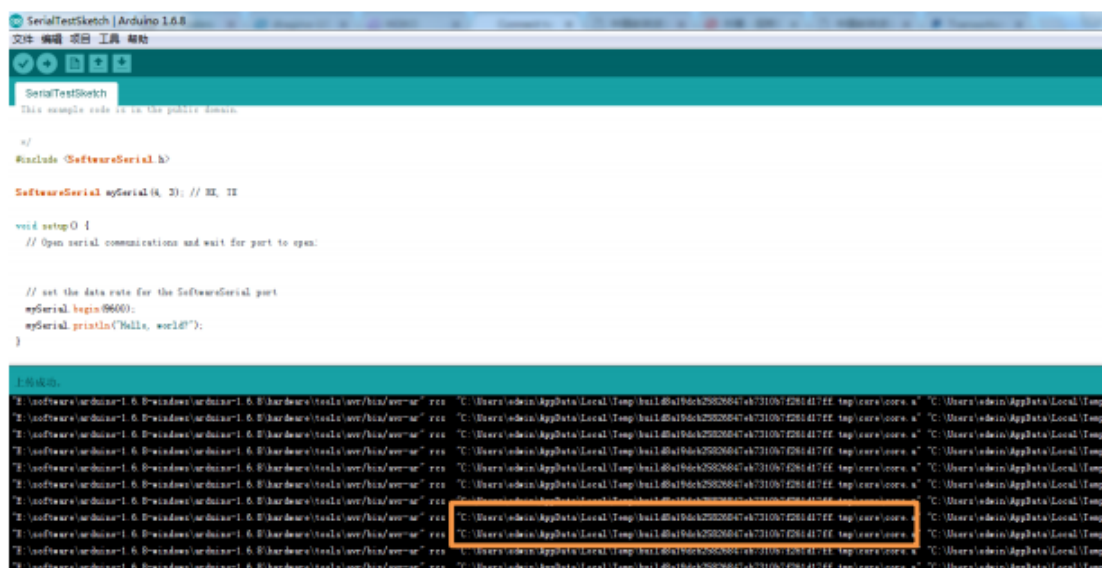
8.2 Web UI によるアップグレード

hex フォーマットに変換されたファイルを使う場合に有用です。

- Web UI で LG01 にアクセスし、**Sensor → Flash MCU** を開き、hex ファイルを選択してアップロードします。
- LG01 を再起動し、**Sensor → MicroController** から MCU のバージョンが更新されているか確認します。

hex ファイルの取得方法

Arduino IDE でコンパイルする際に、コードの配置場所を確認できます。



ディレクトリを開いて 16 進数のファイルを取得します。（ブートローダでは使用しないでください）



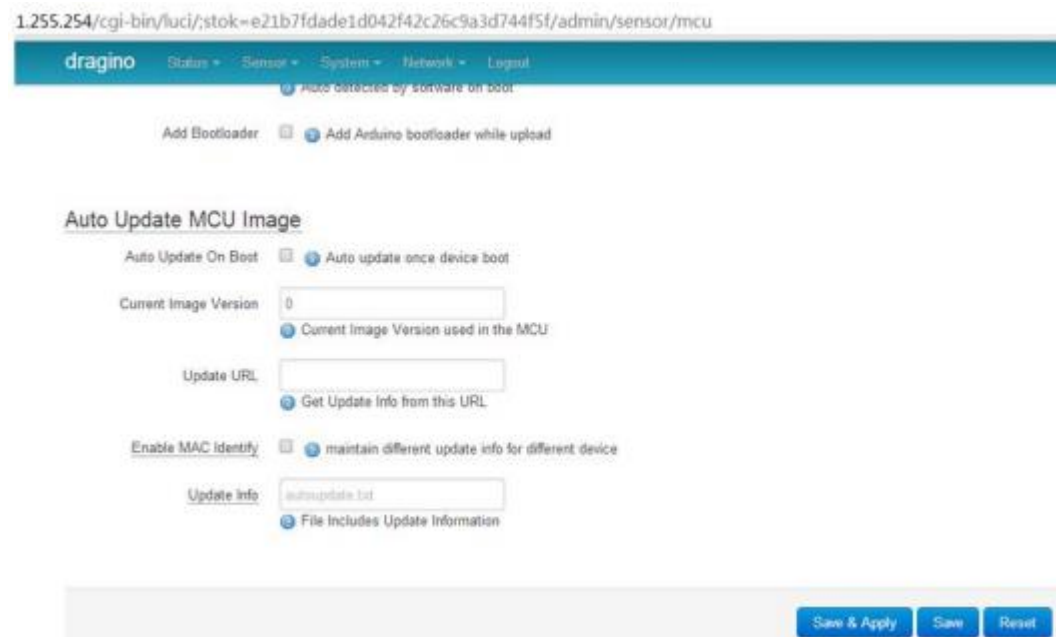
8.3 MCU の自動アップデート

ファームウェア IoT-4.1.1 以降、スケッチの自動更新がサポートされています。この機能により、Dragino の http / https サーバーに接続し、最新のスケッチバージョンを入手してアップロードします。

この機能の目的は、リモートインストールの技術サポートコスト/時間を削減することです。

この機能は、次のページから設定できます。

[Sensors -> microcontroller](#)



Auto Update On Boot: このオプションは有効になっています。 デバイスはすべての起動時に自動サーバーに接続し、更新するスケッチの新しいバージョンがあるかどうかを確認します。 デバイスが自動アップデートサーバー上で新しいバージョンを見つけたら、デバイスはそれをサーバーからダウンロードし、この新しいバージョンで mcu を更新します。

Current Image Version: 現在のスケッチバージョンを表示します。 デフォルトでは 0 です。 デバイスは自動更新が成功した後でのみ、このバージョンを最新のバージョン番号に更新します。

Update URL: この URL は、更新情報と sketch.hex ファイルに問い合わせます。 デバイスはこの URL に接続して、サーバーに新しいバージョンがあるかどうかを確認します。

Update Info: テキストファイルには更新情報が含まれています。下記サイトからファイルの例を取得できます。

http://www.dragino.com/downloads/downloads/tmp/autoupdate/update_info

update information file は以下を含むべきです：

image: アップデートするスケッチ

md5sum: スケッチの md5sum

version: 最新バージョン番号

Enable MAC Identify: アップデート情報で指定されたアップデート情報を取得する代わりに、デバイスはファイル wifi_mac.txt からアップデート情報を探します。つまり、デバイスに Wi-Fi MAC アドレス A8:40:41:78:67:AF がある場合、デバイスはファイルをダウンロードします。自動アップデート情報の場合は \$ Update_URL / A840417867AF.txt です。

自動更新スケッチの手順:

次の構成の場合：

Auto Update On Boot : checked

Update URL: <http://www.dragino.com/downloads/downloads/tmp/autoupdate/>

Update Info: update_info

Enable MAC Identify: unchecked

再起動後、次のようにアップデートされます：

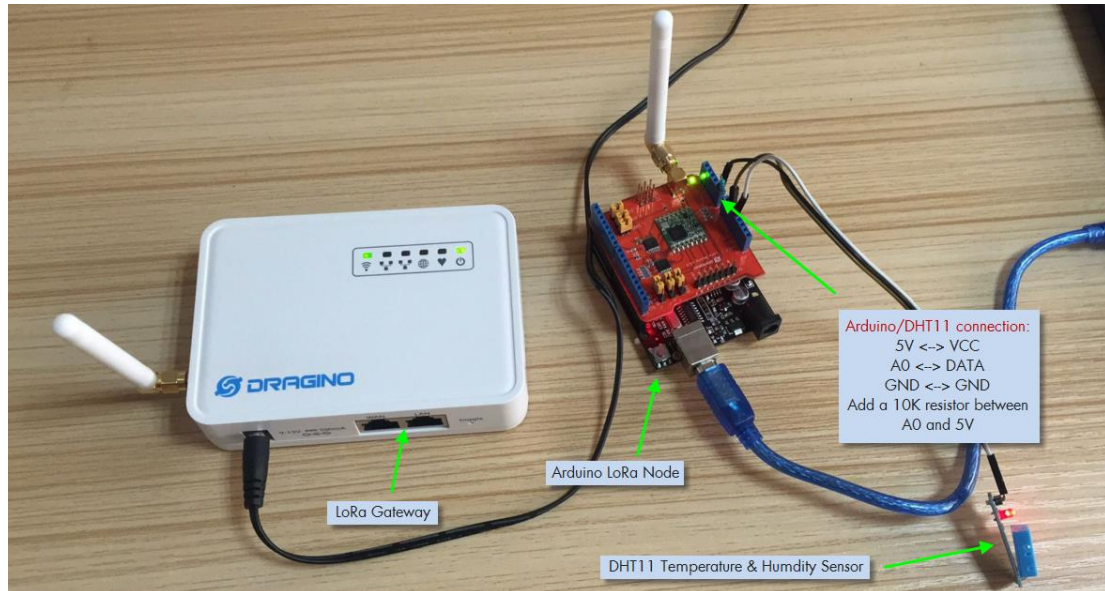
1. アップデート情報を URL から取得します。
http://www.dragino.com/downloads/downloads/tmp/autoupdate/update_info
2. 最新バージョンとデバイス上のバージョンを比較します。
3. サーバー側のバージョンが新しければ、スケッチをダウンロードします。
<http://www.dragino.com/downloads/downloads/tmp/autoupdate/sketch.hex>
4. md5sum チェックを行います。
5. MCU をアップデートします。
6. バージョン番号を最新に更新します。

9. アドバンス事例

9.1 アップストリーム: LoRa ノードからデータ取得し、IoT サーバーへ送信

9.1.1 ハードウェア設定

下記写真のようにハードウェアを設定します LoRa クライアントノードは、温度湿度センサーDHT11 からデータを取得し、そのデータは LoRa 無線で LG01 ゲートウェイに送信されます。LG01 がデータを受信した時にデータは IoT サーバーへ TCP/IP 転送されます。



9.1.2 IoT サーバーアカウントの設定

今回は、IoT サーバーサービスとして [ThingSpeak](https://thingspeak.com/) を取り上げます。

<https://thingspeak.com/>

最初に ThingSpeak でアカウントを作成します。そしてチャンネルに行き、温度と湿度フィールドのチャンネルを作成します。

<https://thingspeak.com/channels/new>


New Channel

Name:

Description:

Field 1: ☒

Field 2: ☒

Field 3: ☐

Field 4: ☐

Field 5: ☐

Field 6: ☐

Help

Channels store all the data that a ThingSpeak application collects. Each channel has eight fields that can hold any type of data, plus three fields for status data. Once you collect data in a channel, you can use ThingSpeak to visualize it.

Channel Settings

- **Channel Name:** Enter a unique name for the ThingSpeak channel.
- **Description:** Enter a description of the ThingSpeak channel.
- **Field#:** Check the box to enable the field, and enter a field name. A channel can have up to 8 fields.
- **Metadata:** Enter information about channel data, including units, precision, and so on.
- **Tags:** Enter keywords that identify the channel. Separate tags with commas.
- **Latitude:** Specify the position of the sensor or thing that is sending data. For example, the latitude of the city of London.
- **Longitude:** Specify the position of the sensor or thing that is sending data. For example, the longitude of the city of London.
- **Elevation:** Specify the position of the sensor or thing that is sending data.

チャンネルを作成した後で、ユーザーは、チャンネルページで **Channel ID** と **API key**を確認することができます。アップデートされたデータは、このチャンネルに取り込まれます。

9.1.3 スケッチをアップロード

Step 1: 下記 URL より [ThingSpeak Library](https://github.com/mathworks/thingspeak-arduino) をダウンロードし、Arduino ライブラリディレクトリに置きます。

<https://github.com/mathworks/thingspeak-arduino>

Step 2: 下記メニューよりサンプルを選択します：

➤ **IDE → File → Examples → Dragino → IoTServer → ThingSpeak → dht11_client**

そしてこのスケッチを LoRa ノードにアップロードします。

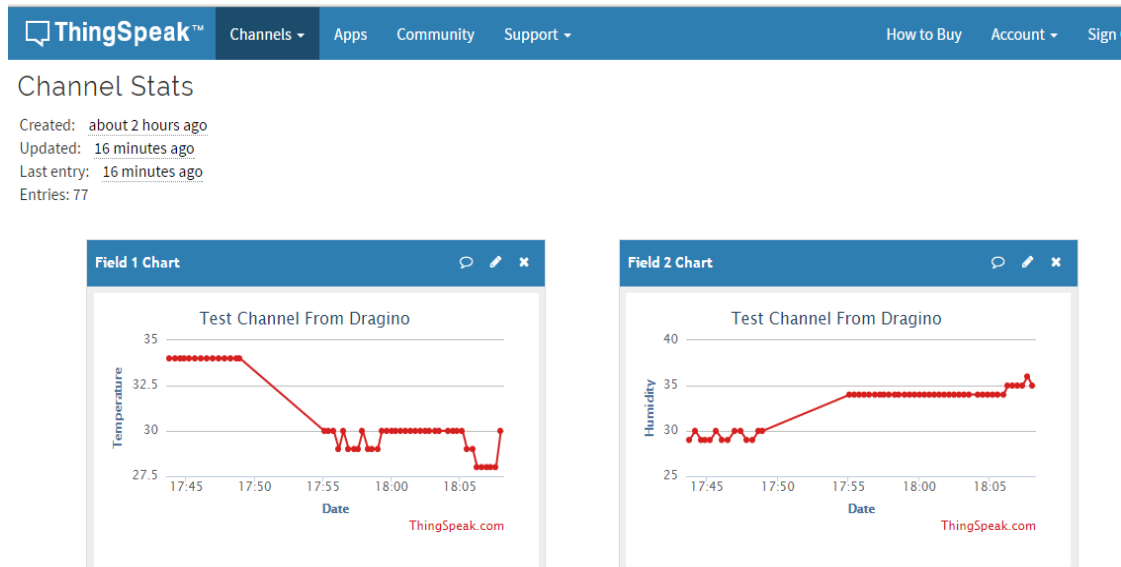
Step 3: 下記メニューよりサンプルを選択します：

➤ **IDE → File → Examples → Dragino → IoTServer → ThingSpeak → dht11_server**

myChannelNumber & myWriteAPIKey をあなたの情報に置き換えます。そしてこのスケッチを LG01 ゲートウェイにアップロードします。

9.1.4 検証結果

ThingSpeak の温度と湿度データの結果は下記のとおり表示されます。



9.2 ダウンストリーム: IoT サーバーからデータ取得し、LoRa ノードに送信

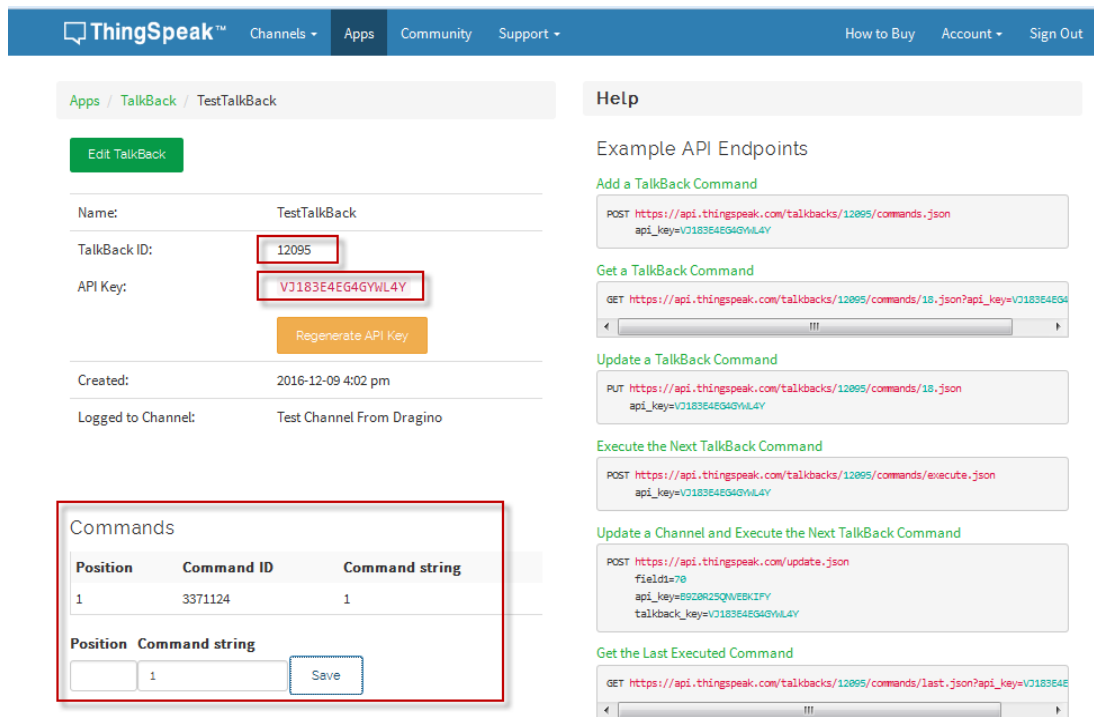
9.2.1 ハードウェア設定

下記写真のようにハードウェアを設定します。LG01 ゲートウェイは、IoT サーバーからコマンドを取得します。そしてコマンドを LoRa クライアントノードへ送信します。受信コマンドに従い、LoRa ノードは接続されたリレーを制御します。



9.2.2 talkback コマンド設定

ThingSpeak アプリで talkback コマンドを作成します。それから、talkback ID と talkback API key を下記画面で確認することができます：



The screenshot displays the ThingSpeak 'TestTalkBack' configuration page. The 'TalkBack ID' is 12095 and the 'API Key' is VJ183E4EG4GYWL4Y. A 'Commands' table is shown with the following data:

Position	Command ID	Command string
1	3371124	1

The right sidebar lists several API endpoints for managing TalkBack commands, such as 'Add a TalkBack Command' (POST /talkbacks/12095/commands.json) and 'Execute the Next TalkBack Command' (POST /talkbacks/12095/commands/execute.json).

9.2.3 スケッチをアップロード Upload sketches

Step 1: 下記でサンプルを選択します：

- IDE → File → Examples → Dragino → IoTServer → ThingSpeak → DownstreamClient

このスケッチを LoRa ノードにアップロードします。

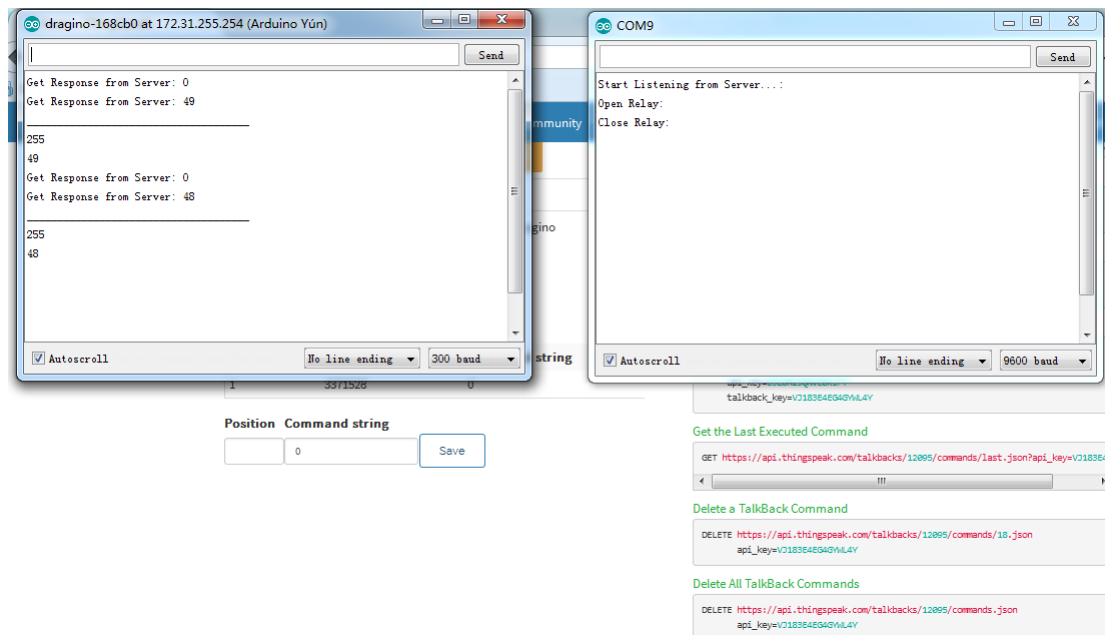
Step 2: 下記でサンプルを選択します：

- IDE → File → Examples → Dragino → IoTServer → ThingSpeak → DownstreamServer

Replace the talkback ID & talkbackAPIKey with your info And upload this sketch to the LoRa Gateway.

9.2.4 検証結果

talkback APP ページでコマンド 1 または 0 でリレーをトリガーします。コンソール画面で下記の結果が確認表示できます。



9.3 他のサンプル

Dragino は、下記 wiki サイトでサンプル事例をアップデートしています：

http://wiki.dragino.com/index.php?title=Examples_Catalog

10. FAQ

10.1 なぜ、433/868/915/920 バージョンがあるのですか？

無免許の省エネ長距離無線を実現する ISM バンド帯の周波数帯域は、各国により異なります。LoRa 通信チップは広域な周波数をサポートしますが、LoRa 部分を各国に応じて最適化するために異なるバージョンをご提供しています。

10.2 LG01 の LoRa バージョンの周波数は？

LoRa 無線チップは下記のようなバージョンで分かれています。特に日本市場向けの LG01-JP は、920MHz をサポートし、TELEC 技適申請済のバージョンを特別に用意しております：

Version	LoRa IC	Support Frequency	Best Tune Frequency
433	Semtech SX1278	Band2 (LF) : 410 ~525MHz Band3 (LF) : 137 ~175MHz	433MHz
868	Semtech SX1276	Band1 (HF) : 862 ~1020MHz	868MHz
915	Semtech SX1276	Band1 (HF) : 862 ~1020MHz	915MHz
920	Semtech SX1276	Band1 (HF) : 862 ~1020MHz	920MHz

10.3 どのような種類の LoRa デバイスがゲートウェイでサポートできますか？

LoRa 部品ソフトウェアは、Mega328P MCU で動作しています。例として Radiohead Library を使用しています。他の LoRa デバイスが同じ Radiohead ライブラリを実行していて、同じ周波数と同じ暗号化を行っている場合、このゲートウェイと通信できるはずです。

MCU 上で他の LoRa プロトコルを実行して、必要な他の LoRa デバイスをサポートすることもできます。

RN2483 をサポートする方法を示す例を以下に示します。

http://wiki.dragino.com/index.php?title=Communicate_with_RN2483

10.4 LG01 はいくつのノードをサポートできますか？

最大サポートエンドノードは、エンドノードとゲートウェイ間の通信（頻度）に依存します。単純な LoRa の例を使用した研究室のテストでは、エンドノードが 5 分ごとにゲートウェイにデータを送信しようとする、チャンネルコリジョンのためにネットワークに 20~30 のノードがあるとデータが失われます。

ユーザーがより多くのノードにアクセスしたい場合、ユーザーはポーリング方法を使用することを検討してください。ゲートウェイがポーリング方法を使用してエンドノードからデータを取得する場合、ゲートウェイは数百ノード以上をサポートできます。例には次のものがあります。

<https://github.com/dragino/Arduino-Profile->

<Examples/tree/master/libraries/Dragino/examples/LoRa/Concurrent>

10.5 LG01 がサポートできるサーバーの種類は？

LG01 の Linux 側は OpenWrt です。オープンソースであり、ユーザーはアプリケーションを開発することができます。

基本的には、適切な API を使用すればほとんどの IoT サーバーをサポートできます。IoT、MQTT、または RESTful のための典型的なプロトコル（MQTT、RESTful）を介していくつかのサーバーを接続する方法の例があります。

<https://github.com/dragino/Arduino-Profile->

<Examples/tree/master/libraries/Dragino/examples/IoTServer>

10.6 LG01 のために私自身のファームウェアを作ることができますか？ このソースコードを見つけることができますか？

はい、独自のブランド化したり、カスタマイズアプリを追加したりして、LG01 オリジナルファームウェアを開発することはできます。LG01 のソースコードは下記で公開されています。そしてインストラクションに従い、コンパイルして再利用できます：

<https://github.com/dragino/openwrt-cc-15.05>

10.7 このデバイスのサンプルをもっと入手するには？

Arduino IDE の Dragino examples ディレクトリに Arduino のサンプルを公開しています。ユーザーが先に Dragino ボードをインストールして、新しいサンプルをリリースした場合、ユーザーがボードプロファイルを更新する以外は、新しいものは IDE に表示されません。更新するには、ユーザーは Arduino ボードマネージャーでボードプロファイルを削除し、再度インストールすることができます。

10.8 一般的な LoRa の質問に関するその他のよくある質問について

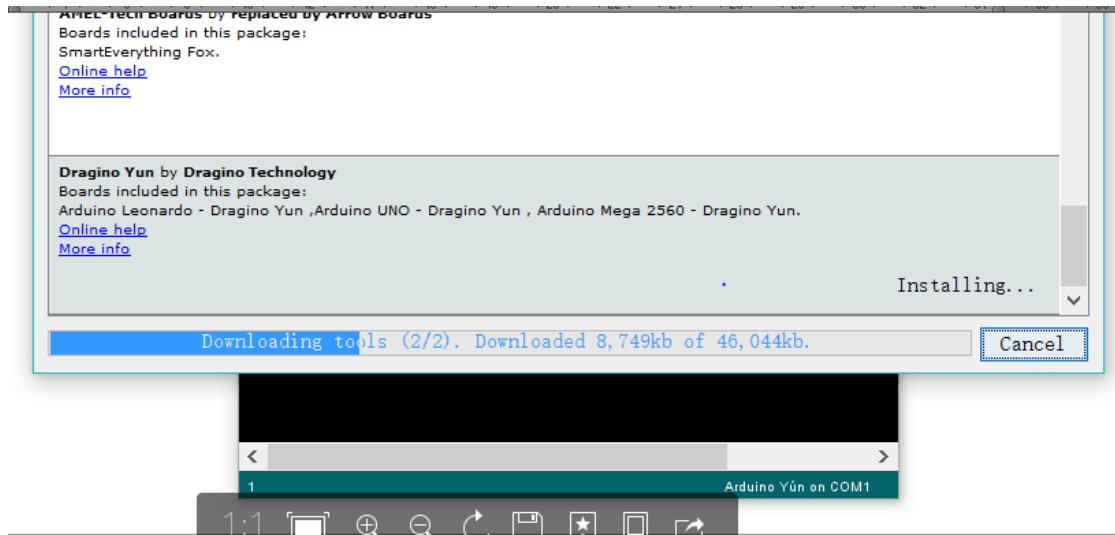
Wiki でよくある質問を更新しています。

http://wiki.dragino.com/index.php?title=LoRa_Questions

11. トラブルシューティング:

11.1 Dragino プロファイルをダウンロードできません

Board manager で Dragino プロファイルをダウンロードしているにもかかわらず、Arduino IDE がまったく反応していない場合、どこかに行方不明になっているかもしれません。下記のような場合には、Arduino IDE からパッケージをインストールするネットワーク環境に問題があり、反応が遅い状態が考えられます。



こういう問題を解決するには、ユーザーはマニュアルで下記のステップで Dragino プロファイルを追加することができます:

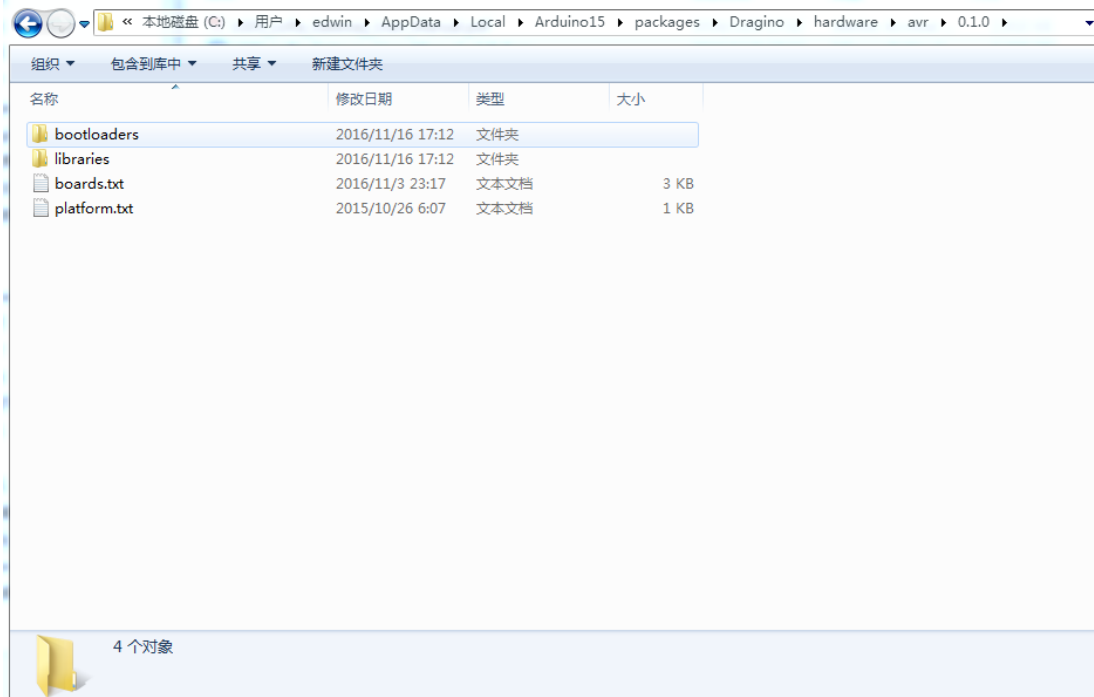
STEP 1: 下記 URL からプロファイルをダウンロードします。

<https://github.com/dragino/Arduino-Profile-Examples>

STEP 2: ダウンロードしたファイルを解凍し下記ディレクトリにファイルを置きます。

C:\Users\%xxx%\AppData\Local\Arduino15\packages\Dragino\hardware\avr\0.1.0

注記: 異なるシステムでは、Arduino15 の異なるディレクトリ構造となる場合があります。もし、Dragino\hardware\avr\0.1.0 を見つけることができなければ、Arduino15 ディレクトリを作成してください。最終的なディレクトリ構造は下記画面キャプチャーのようになります。



11.2 MCU と Linux システムとのブリッジが動作しません

いくつかの問題の可能性として：

SETP 1: ブリッジは、MCU スケッチの **Serial class** を使います。Serial.begin(9600) のような MCU スケッチでは、Mega328P のブリッジライブラリは、同じシリアルインターフェースを使用します。スケッチのシリアルコードが同じ場合、それらは衝突して動作することができません。

STEP 2: 他のスケッチをコンパイルした場合、Arduino IDE はシリアル設定で混乱を起こします。このような場合、一度 Arduino IDE を終了して再度起動してみてください。

11.3 Arduino IDE が LG01 を認識できません

もし問題が発生したら、下記項目をチェックしてください：

- ✓ Arduino IDE のバージョンが 1.5.4 またはそれ以降
- ✓ あなたのパソコンが、Yun LG01 と同じネットワークにあるかどうか
- ✓ ウェブまたは SSH 経由で LG01 にアクセスし、IDE を再度確認
- ✓ 上記でも認識できない場合、LG01 に SSH ログインし、次のコマンドを実行します
/etc/init.d/avahi-daemon restart

これは、IDE から LG01 を認識するためのプロセスの再起動です

11.4 新しいパッケージをインストールするときにカーネルエラーが発生した場合の修正方法は？

次のようなカーネルエラーが発生したとします。

```
root@dragino-16c538:~# opkg install kmod-dragino2-si3217x_3.10.49+0.2-1_ar71xx.ipk

Installing kmod-dragino2-si3217x (3.10.49+0.2-1) to root...

Collected errors:

* satisfy_dependencies_for: Cannot satisfy the following dependencies for kmod-dragino2-si3217x:

* kernel (= 3.10.49-1-4917516478a753314254643facdf360a) *

* opkg_install_cmd: Cannot install package kmod-dragino2-si3217x.
```

こういったエラーの場合、`-force-depends` オプションを指定してください。

```
opkg install kmod-dragino2-si3217x_3.10.49+0.2-1_ar71xx.ipk --force-depends
```

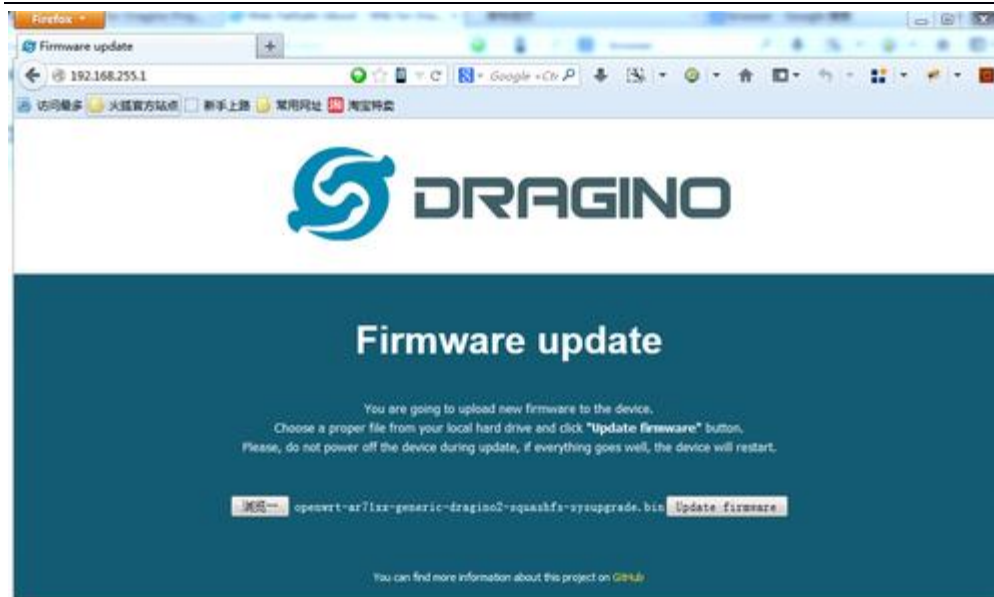
11.5 ファームウェアがクラッシュした場合、どうやって LG01 をどう回復できますか？

LG01 は、Linux システムでユーザーにフルコントロールできます。ブート可能なファイルの一部が不適切に変更された場合、デバイスがブリックして起動できなくなる可能性があります。

このような場合、ユーザーは、ウェブファイルセーフモード経由で新しいファームウェアをアップロードすることで全体の Linux システムを回復することができます。

手順は下記のとおりです：

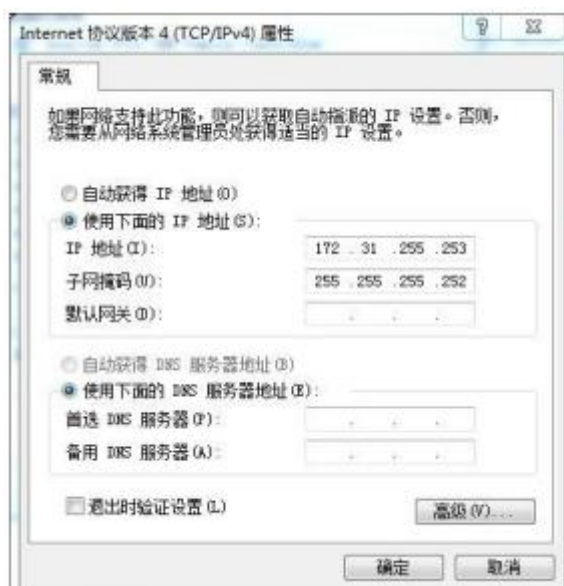
- RJ45 ケーブルを使い、パソコンと LG01 の RG45 ポートを接続します。
- パソコンから IP アドレスを 192.168.255.x, サブネットマスク 255.255.255.0 に設定します。
- IP01 のトグルリセットボタンを 30 秒以上押しながら、IP01 の電源をオンにします。
- IP01 の全ての LED が点滅します。4 つの LED が点滅した後でトグルリセットボタンを離します。
- IP01 の全ての LED がすばやく一度点滅します。これはネットワーク接続を認識したという意味です。そしてウェブファイルセーフモードに入ります。IP01 がファイルセーフモードに入った後で、パソコンで PING コマンド 192.168.255.1 を実行します。
- パソコンのウェブブラウザで 192.168.255.1 を開きます。
- [squashfs-sysupgrade](#) 入力し、下記画面が右下 update firmware ボタンを押します。



11.6 WiFi アクセス用に LG01 を設定して IP を失った場合にどうすればよいか
LG01 は LAN ポートにフォールバック IP を持っています。この IP は常に有効になっているため、WiFi IP に関係なく LG01 にアクセスするためにフォールバック IP を使用できます。フォールバック IP は、ユニットの接続とデバッグに便利です。

注意：フォールバック IP は LAN and DHCP ページで無効にできます。

- 1.イーサネット LAN ポートで LG01 に接続します。
2. PC 側の設定を IP: 172.31.255.253 、ネットマスク: 255.255.255.252 に背ってします。



3. 172.31.255.254 で LG01 に Web かコンソールでアクセスします。

11.7 Linux パートの工場初期化をするには？

「6.1 ネットワークリセット又は工場規定値にリセット」の手順は、MCU パートのリセットです。Linux パートのリセットは次の手順でおこないます。（「4.1Linux コンソール用 SSH アクセス」を参考に、Linux コンソール上で実行します。）

Linux コンソールを実行：

```
root@dragino-169d30:~# mtd -r erase rootfs_data // リセットコマンド実行
```

```
Unlocking rootfs_data ...
```

```
Erasing rootfs_data ...
```

```
Rebooting ...
```

Linux のシャットダウンが実行され、電源 LED のみ点灯する状態になります。

数分経過してもインターネット LED の点滅が始まらない場合、AC 電源を指し直し、インターネット LED の点滅が始まることを確認します。

再度 SSH で Linux コンソールが開ければ、リセットの完了です。

12. 注文情報

- **LG01P-433**：LoRa 無線 433 MHz に LG01 ゲートウェイを最適化
- **LG01P-868**：LoRa 無線 868 MHz に LG01 ゲートウェイを最適化
- **LG01P-915**：LoRa 無線 915 MHz に LG01 ゲートウェイを最適化
- **LG01-JP (920)**：日本国内市場向け 920 MHz に LG01 ゲートウェイを最適化

13. 梱包内容

商品梱包は下記内容が含まれます：

- ✓ LG01 LoRa ゲートウェイ本体 x 1 台
- ✓ スティックアンテナ（LoRa 無線）
- ✓ 電源アダプター：EU/AU/US タイプの AC アダプター
- ✓ 環境対応の紙ケース箱

外寸と重量：

- ✓ Device Size: 12 x 8.5 x 3 cm
- ✓ Device Weight: 150g
- ✓ Package Size / pcs : 21.5 x 10 x 5 cm
- ✓ Weight / pcs : 360g
- ✓ Carton dimension: 45 x 31 x 34 cm. 36pcs per carton
- ✓ Weight / carton : 12.5 kg

14. 参照

◇ LG01 オフィシャル wiki

http://wiki.dragino.com/index.php?title=Examples_Catalog

- ✧ LG01 LoRa ゲートウェイソースコード
<https://github.com/dragino/openwrt-cc-15.05>
- ✧ ゲートウェイのサンプルプログラム
http://wiki.dragino.com/index.php?title=Examples_Catalog
- ✧ OpenWrt オフィシャル Wiki
<http://www.openwrt.org/>
- ✧ Arduino オフィシャルサイト:
<https://www.arduino.cc>
- ✧ Arduino ブリッジ事例:
<https://www.arduino.cc/en/Tutorial/Bridge>

15. サポート

Dragino 国内総発売元



株式会社オープンウェーブ

〒221-0822 神奈川県横浜市神奈川区西神奈川 1-8-11-103

Tel: 045-620-6648

Email: info@openwave.co.jp

URL: <http://www.openwave.co.jp>

商品企画 & 翻訳



シーアールアイジャパン株式会社

〒107-0062 東京都港区南青山 2-12-15 南青山二丁目ビル 5 階

Tel: 03-4579-5828

Email: info@crijapan.jp

URL: <http://www.crijapan.jp>