



LG01 LoRa Gateway User Manual Document Version: 1.0

Firmware Version: IoT Mesh v3.4.0

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	イントロダクション	4
1.1	概要	4
1.2	LG01 とは	4
1.3	仕様	4
1.4	特長	6
1.5	システム構成	6
1.6	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	プログラムマイクロコントローラー	9
2.2.1	Arduino IDE の設定	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 無線サンプル	13
2.3.1	LoRa ライブラリーをインストール	13
2.3.2	LoRa サーバーにスケッチをアップロード	14
2.3.3	LoRa クライアントスケッチ	14
3	典型的なネットワーク設定	15
3.1	概要	15
3.2	典型的な WiFi アクセスポイントネットワーク	16
3.3	WAN ポートインターネットモード	17
3.4	WiFi クライアントモード	17
3.5	WiFi メッシュネットワーク	18
3.5.1	メッシュゲートウェイ設定	18
3.5.2	メッシュクライアント設定	20
3.6	USB ダイアルアップモデム設定	22
3.7	USB 3G/4G イーサーネットドングル	23
4	Linux システム	24
4.1	SSH アクセス (Linux コンソール)	24
4.2	編集とファイル転送	25
4.3	ファイルシステム	25
4.4	パッケージメインテナンスシステム	26
5	ブリッジライブラリー	27
6	アドバンスマネージメント	28
6.1	ネットワークリセット or 工場規定値	28
7	アップグレードファームウェア	29
7.1	Web UI 経由でアップグレード	29
7.2	Linux コンソール経由でアップグレード	29
8	自動プロビジョニング	30
9	アドバンス事例	30
9.1	アップストリーム: LoRa ノードからデータ取得し IoT サーバに送信	30

---

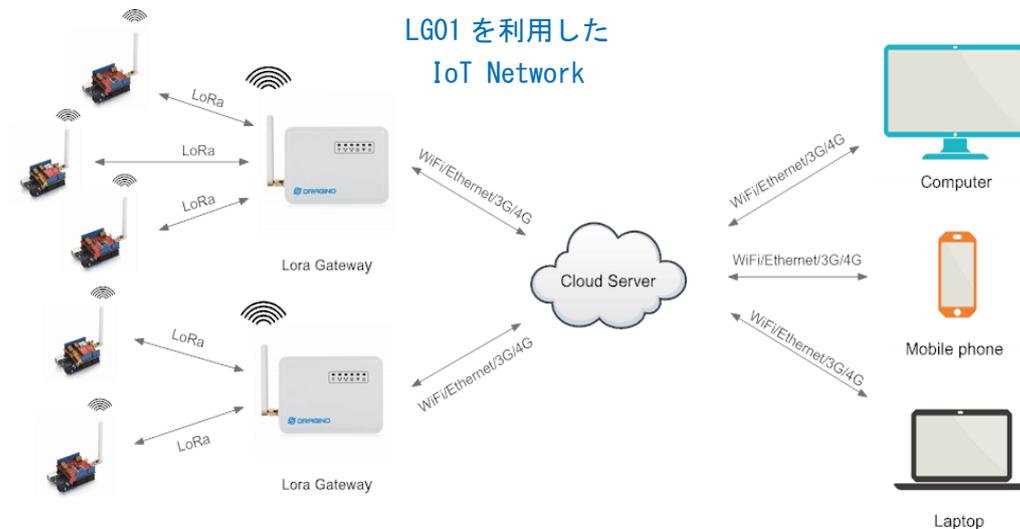
9.1.1	ハードウェア設定.....	30
9.1.2	IoT サーバーアカウント設定 .....	30
9.1.3	スケッチのアップロード.....	31
9.1.4	検証結果.....	31
9.2	ダウンストリーム: IoT サーバーからデータ取得し LoRa ノードに送信 ...	32
9.2.1	ハードウェア設定.....	32
9.2.2	talkback コマンド設定 .....	32
9.2.3	スケッチのアップロード.....	33
9.2.4	検証結果.....	33
9.3	さらに事例.....	34
10	FAQ.....	35
10.1	なぜ、433/868/915 バージョンがあるのですか? .....	35
10.2	LG01 LoRa バージョンの周波数は? .....	35
10.3	自分のファームウェアを利用できますか? LG01 のソースコードはどこにありますか? .....	35
11	トラブルシューティング:.....	36
11.1	Dragino プロファイルをダウンロードできません .....	36
11.2	MCU と Linux モジュールが連動で機能しません.....	37
11.3	Arduino IDE が LG01 を検出できません .....	37
11.4	ファームウェアがクラッシュした時 LG01 をどう回復しますか.....	38
12	注文情報.....	39
13	梱包情報.....	39
14	サポート.....	39
15	参照.....	40

## 1. イントロダクション

### 1.1 概要

#### 1.2 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。



### 1.3 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
  - ✓ Band 3 (LF): 137 ~ 175 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

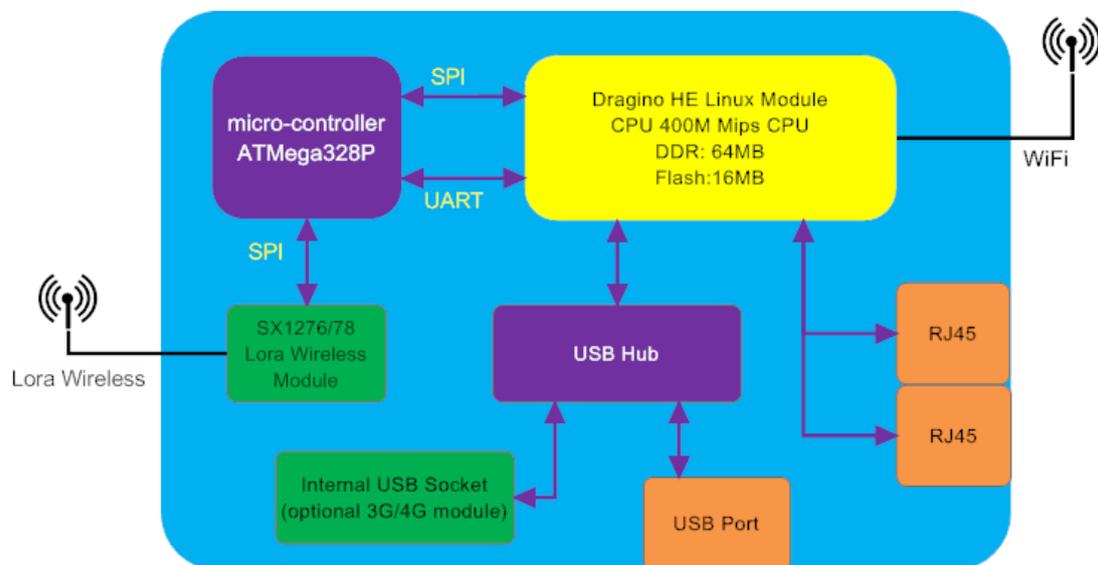
#### 1.4 特長

- ✓ 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

#### 1.5 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 Contraller 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:



1.6 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。

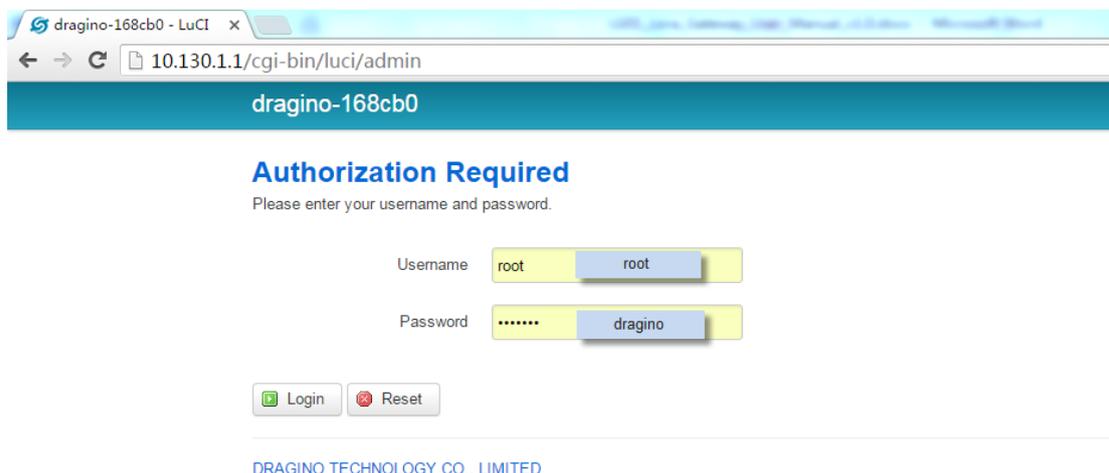
10.130.1.1

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



## 1 マイクロコントローラー・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。マイクロ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。

### 1.1.1 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。

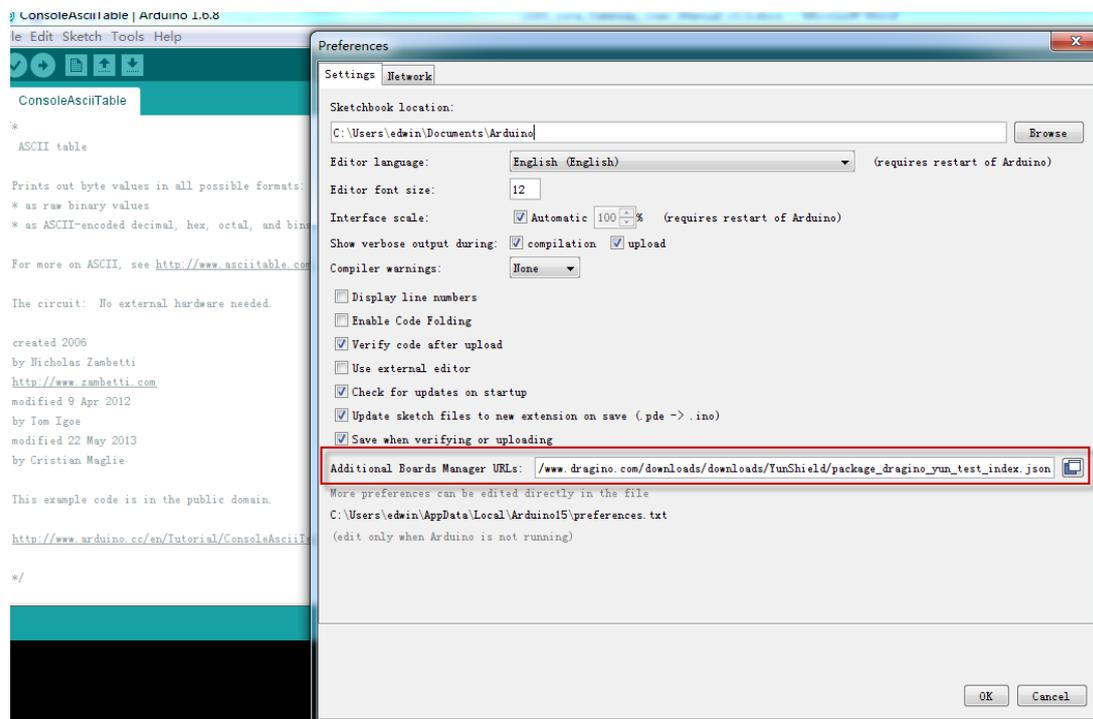
下記が Arduino 公式サイト：

<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、

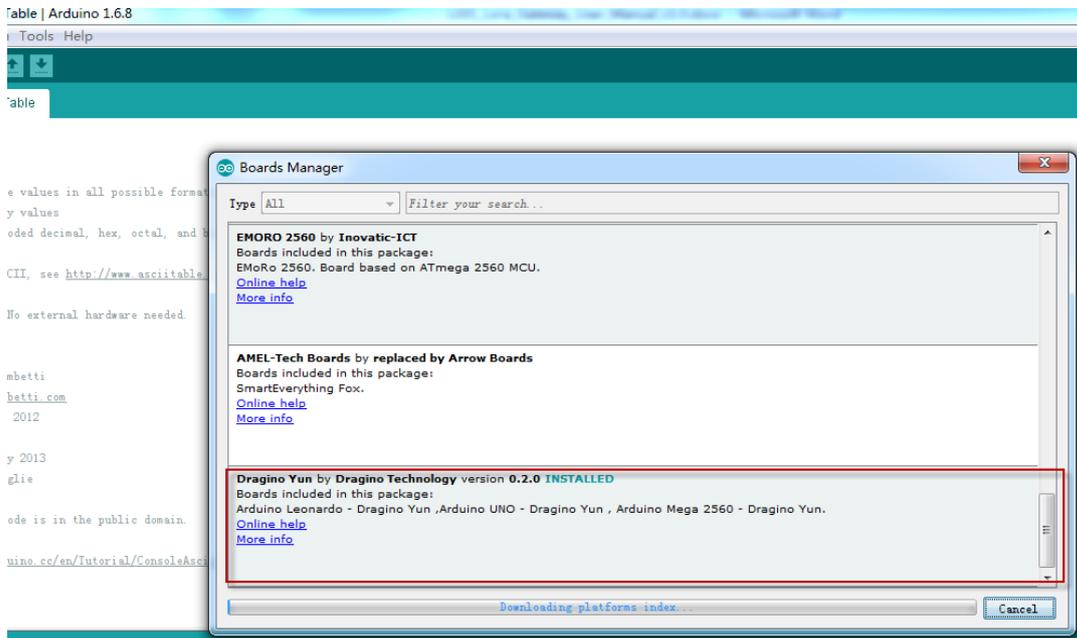
- **File → Preference**, を選択して下記 **Additional Boards Manager URLs** に下記 URL を入力します。

[http://www.dragino.com/downloads/downloads/YunShield/package\\_dragino\\_yun\\_test\\_index.json](http://www.dragino.com/downloads/downloads/YunShield/package_dragino_yun_test_index.json)

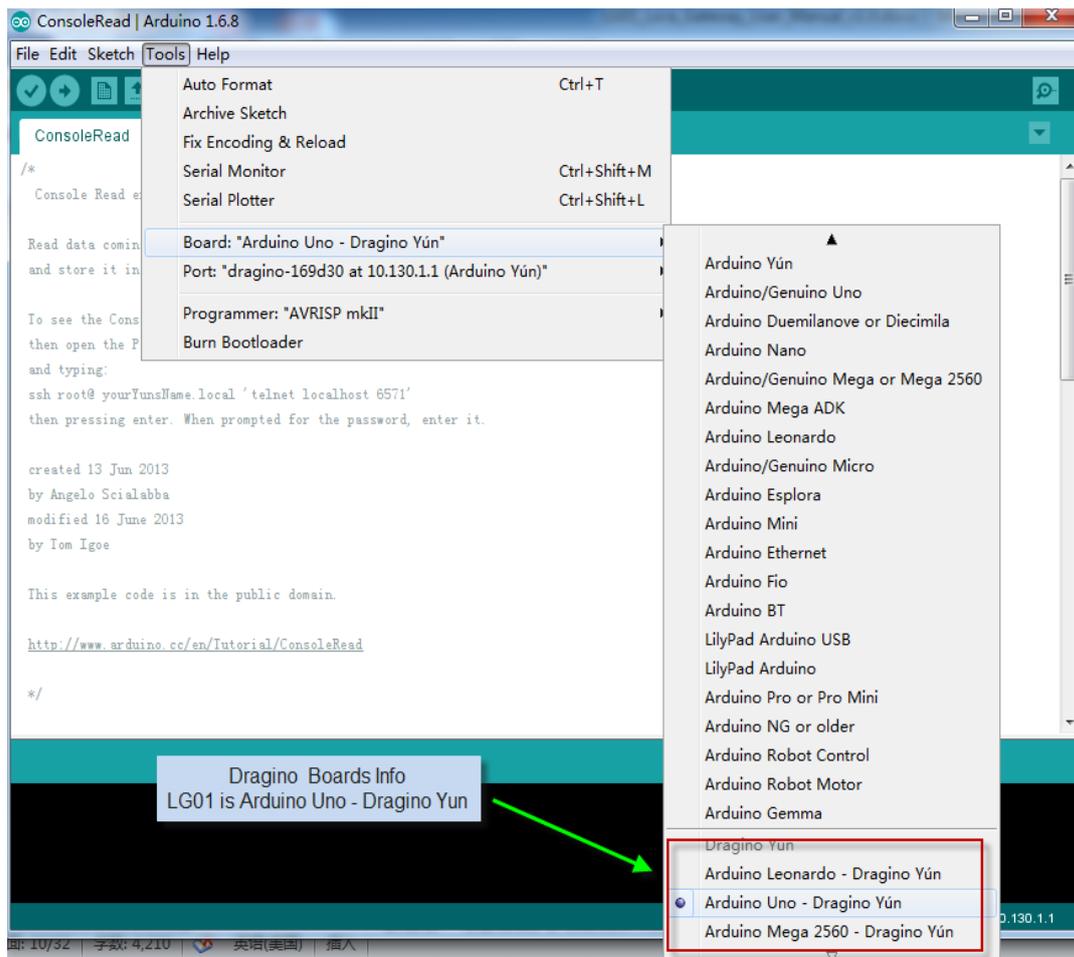


OK ボタンをクリックします。

- **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：
- **Arduino Uno - Dragino Yun**

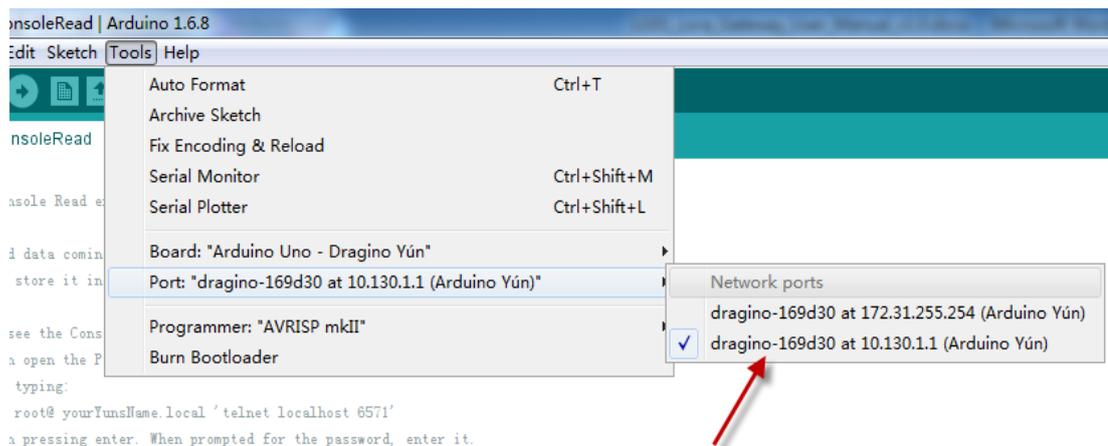


### 1.1.2 スケッチを MCU にアップロード

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

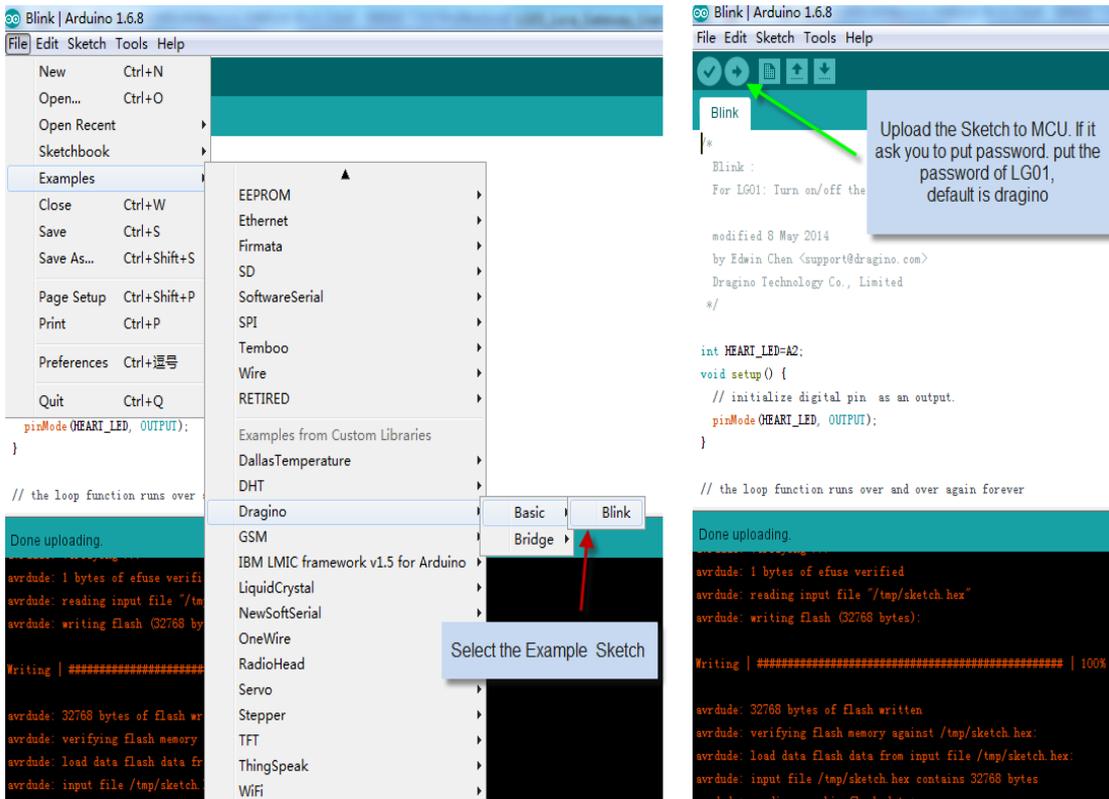
す。Arduino IDE で下記ポート画面で Arduino Yun をチェックします。



Select correct network port in the IDE, we will use this port to program the MCU in LG01

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 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



または

- 右側 LoRa Server：LG01；左側 LoRa Client：LG01

2 つの LoRa 機能ですが、それぞれクライアントとサーバー機能を使い分けることができます。



### 2.1.1 LoRa ライブラリをインストール

Radihead 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。

<http://www.airspayce.com/mikem/arduino/RadioHead/RadioHead-1.63.zip>

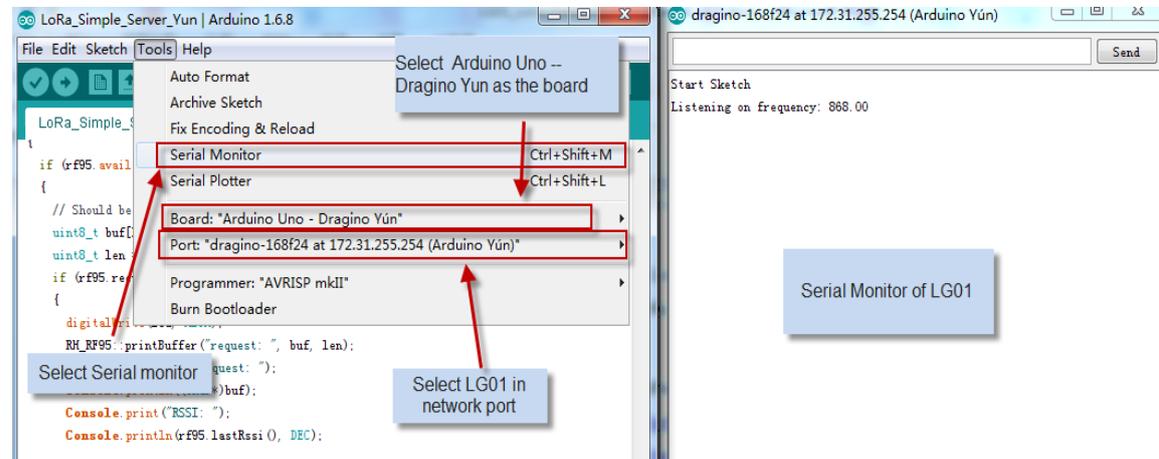
ダウンロード後にファイルを解凍して Arduino ライブラリーフォルダーにインポートします。詳細は下記をご参照ください：

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

### 2.1.2 LoRa サーバースケッチをアップロード

下記メニューを選択します：

- **IDE** → **File** → **Examples** → **Dragino** → **LoRa** → **LoRa\_Simple\_Server\_Yun**
- スケッチを LG01 にアップロードします。LG01 のステータスを確認するためにシリアルモニターメニューを開きます。



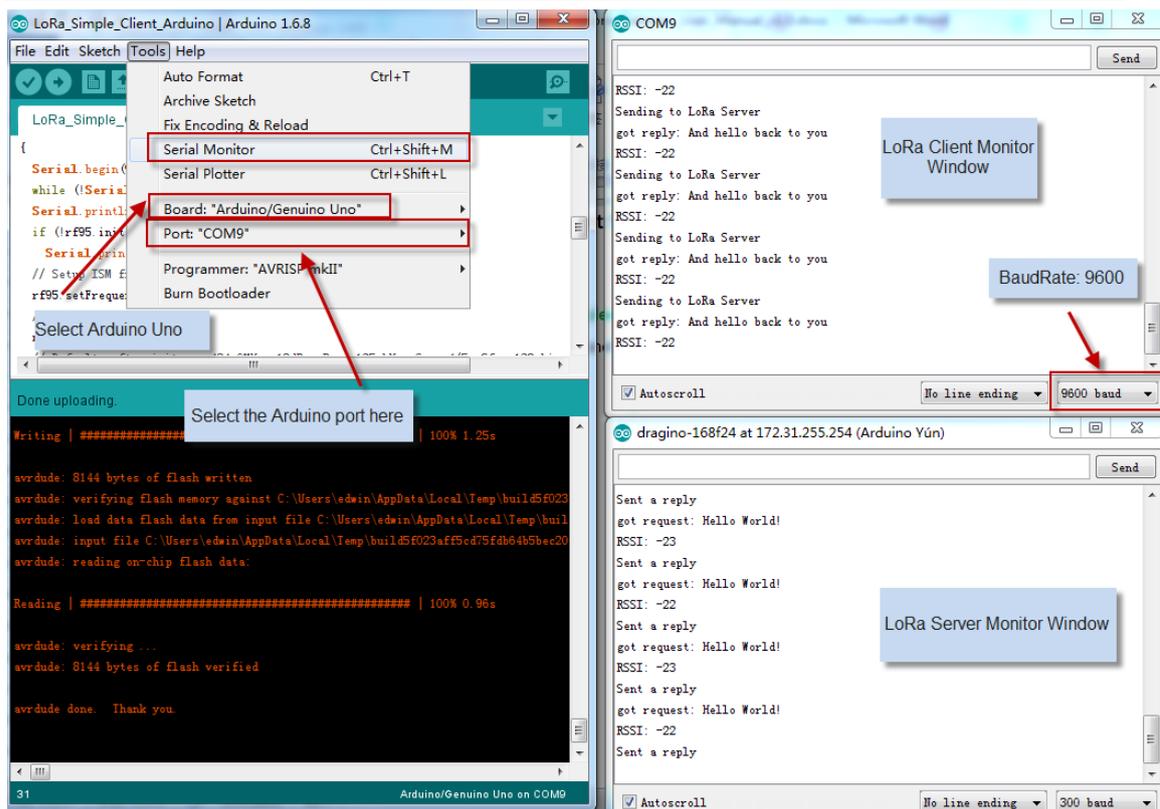
### 2.1.3 LoRa クライアントスケッチをアップロード

LoRa Shield + Arduino を LoRa クライアント設定：

下記メニューを選択します：

- **IDE** → **File** → **Examples** → **Dragino** → **LoRa** → **LoRa\_Simple\_Client\_Arduino**

スケッチを Arduino board にアップロードします。シリアルモニターメニューを開きます。



### 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

#### 4 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 イーサーネットモード

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**

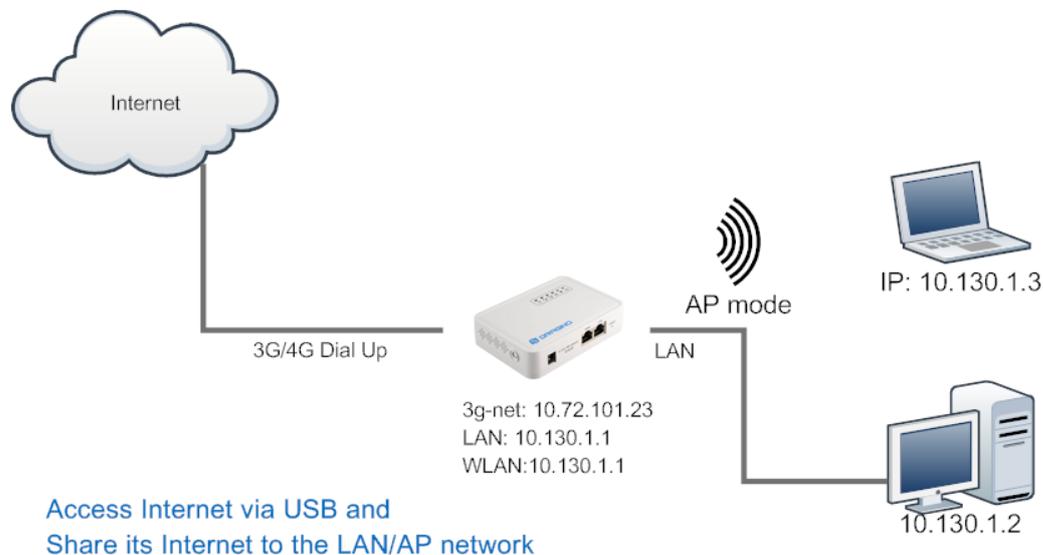
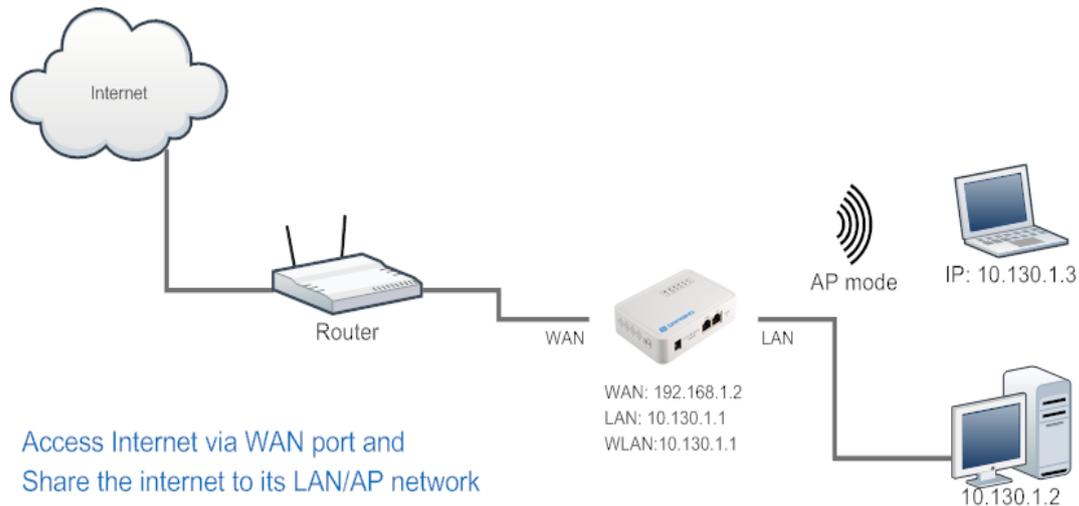
LG01 はパソコンと接続してネットワーク接続を確認することができます。そして **GLOBAL LED** でステータスを確認します。LG01 はホストに正しく接続されている場合は、**GLOBAL LED** が点滅します。



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。

## 5 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

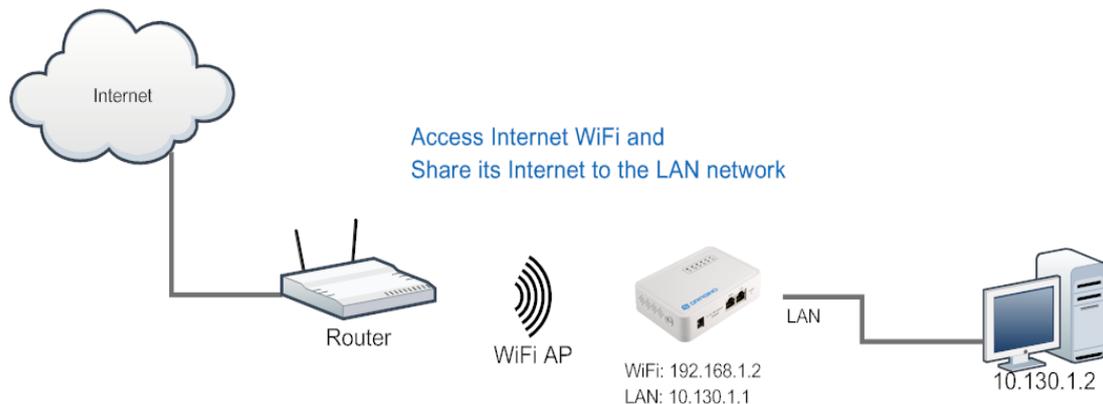
- **Network → Access Point**
- ✓ WiFi アクセスポイントを可能
- ✓ SSID/ Encryption/ パスワードを入力
- 
- **Network → Mesh Network:**
- ✓ WiFi Mesh ネットワークを可能

## 6 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。

## 7 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**

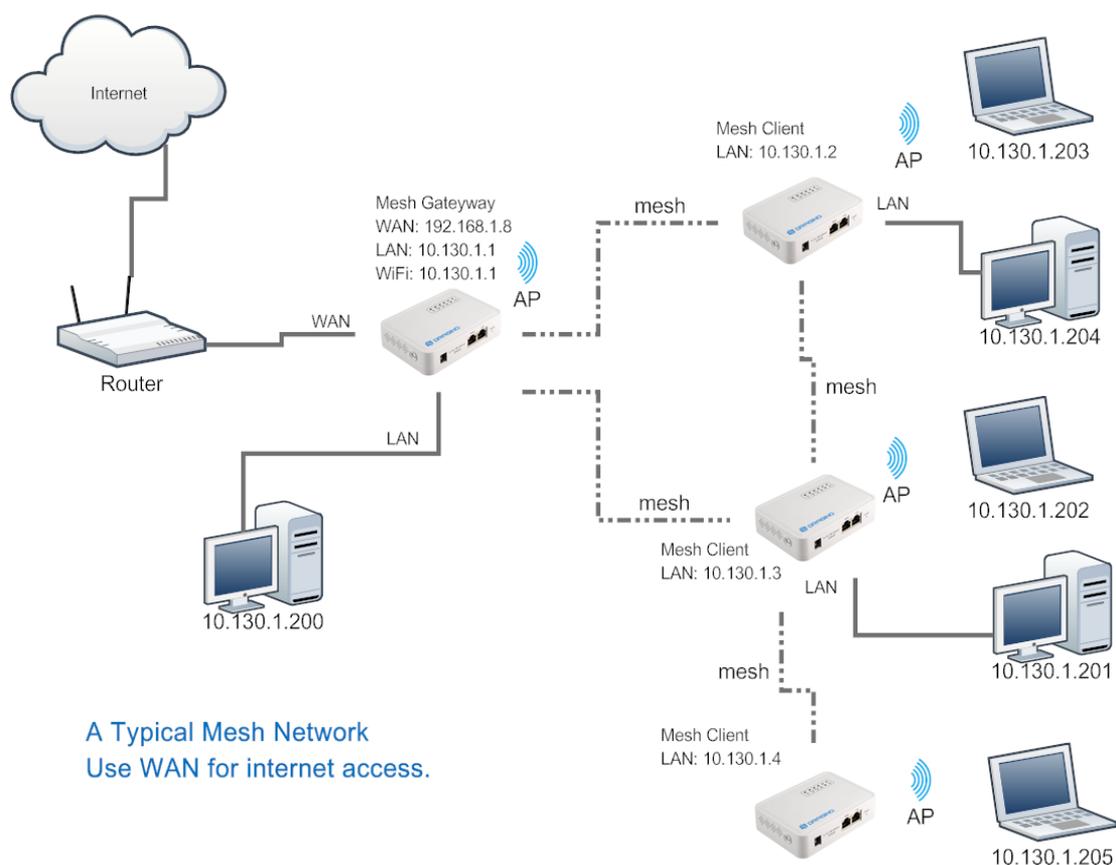
✓ WiFi Mesh ネットワーク可能

## 8 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。



### 8.1.1 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 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

#### 8.1.2 Mesh クライアント設定

- [Network](#) → [Internet Access](#)
- ✓ インタ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 its WAN access, so packets will pass to Mesh Interface.

Display Net Connection

Continuously 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

## Small Enterprise-Campus Network

### LAN and DHCP

IP Address	<input type="text" value="10.130.1.2"/>	Set a unique IP address for its LAN and WiFi interface.
Enable DHCP	<input type="checkbox"/> <input checked="" type="radio"/> Enable DHCP Server	Disable DHCP server in this device.
LAN Gateway	<input type="text" value="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。

## 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

## 9 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。

### 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	<input type="text"/>	Leave blank if no provided by your provider
Password	<input type="password"/>	Leave blank if no provided by your provider
PIN	<input type="text"/>	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

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

**USB Modem Setting**

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  Input USB dongle VID

PID  Input USB dongle PID

Service APN

Dial String  Dial String for Chinatelecom

Username  User Name

Password  Password

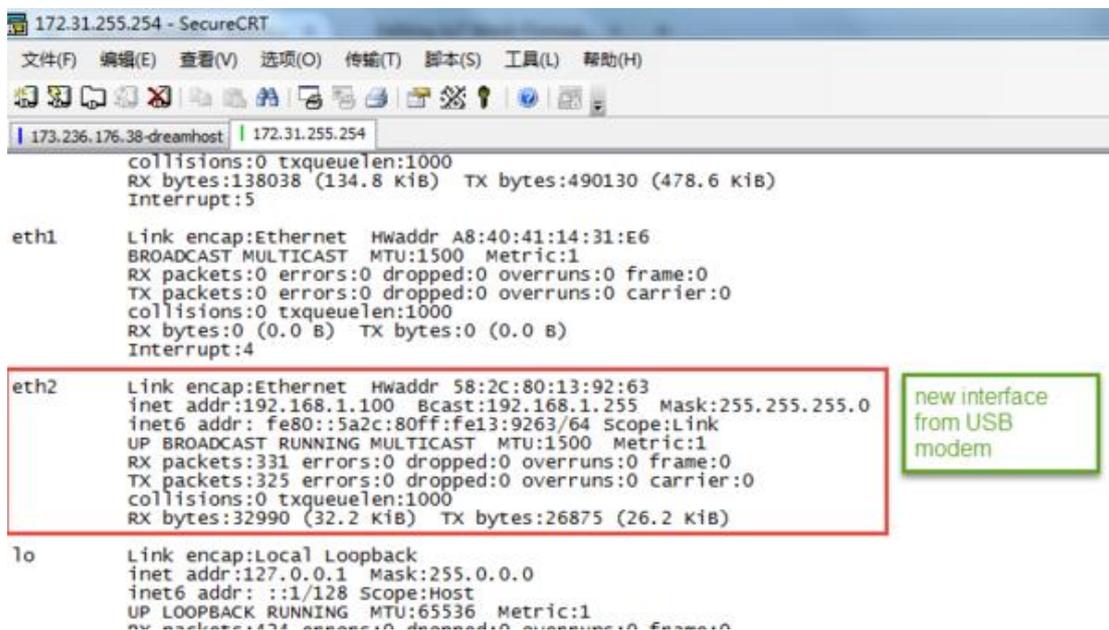
PIN

USB Serial Port ttyUSB0 ▾ Choose USB Serial Port for 3G

Save & Apply

## 9. 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ヤ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。



```

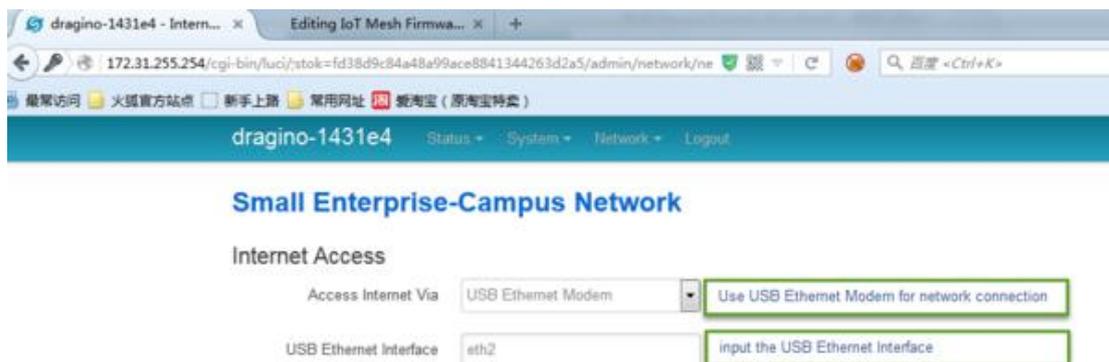
172.31.255.254 - SecureCRT
文件(F) 编辑(E) 查看(V) 选项(O) 传输(T) 脚本(S) 工具(L) 帮助(H)
| 173.236.176.38-dreamhost | 172.31.255.254
collisions:0 txqueuelen:1000
RX bytes:138038 (134.8 KiB) TX bytes:490130 (478.6 KiB)
Interrupt:5

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
        RX packets:134 errors:0 dropped:0 overruns:0 frame:0
    
```

a new interface generate by USB Ethernet modem

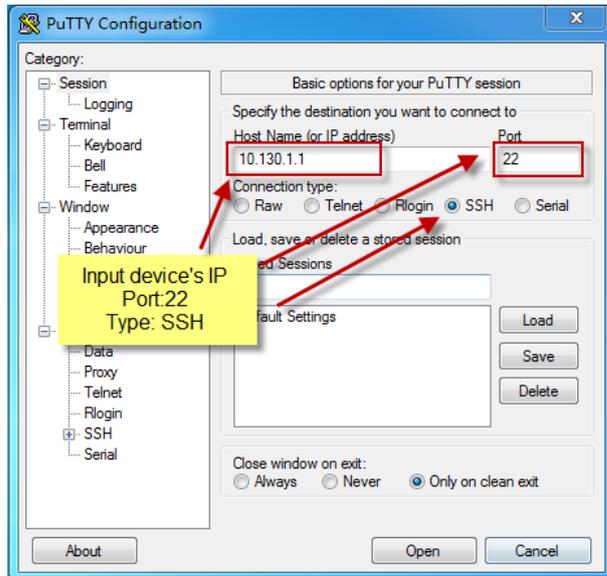


## 10 Linux システム

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。

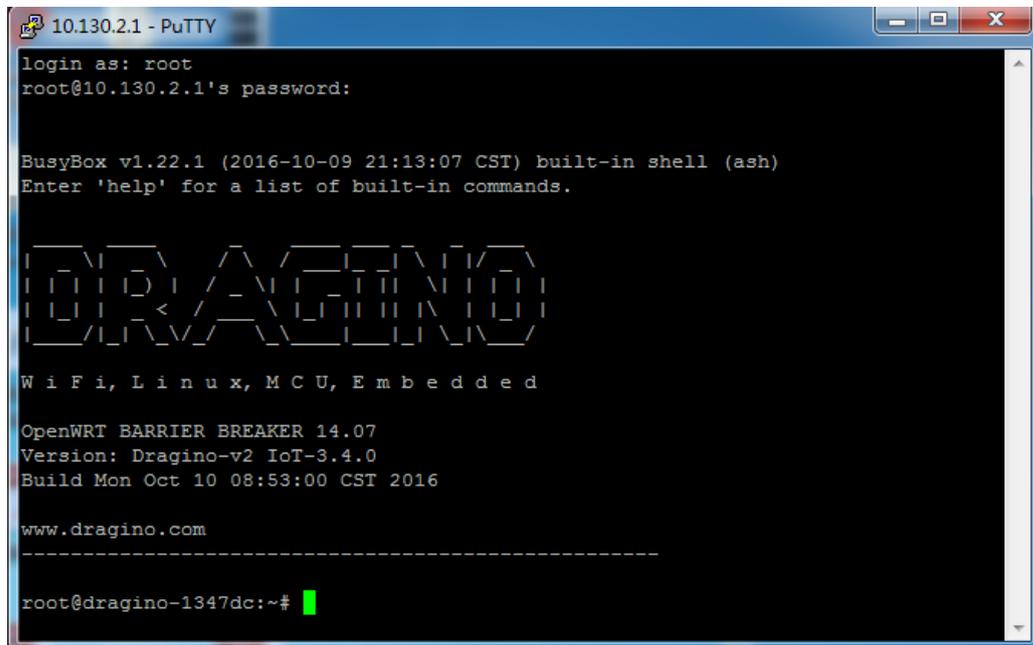
## 11 Linux コンソール用 SSH アクセス

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:



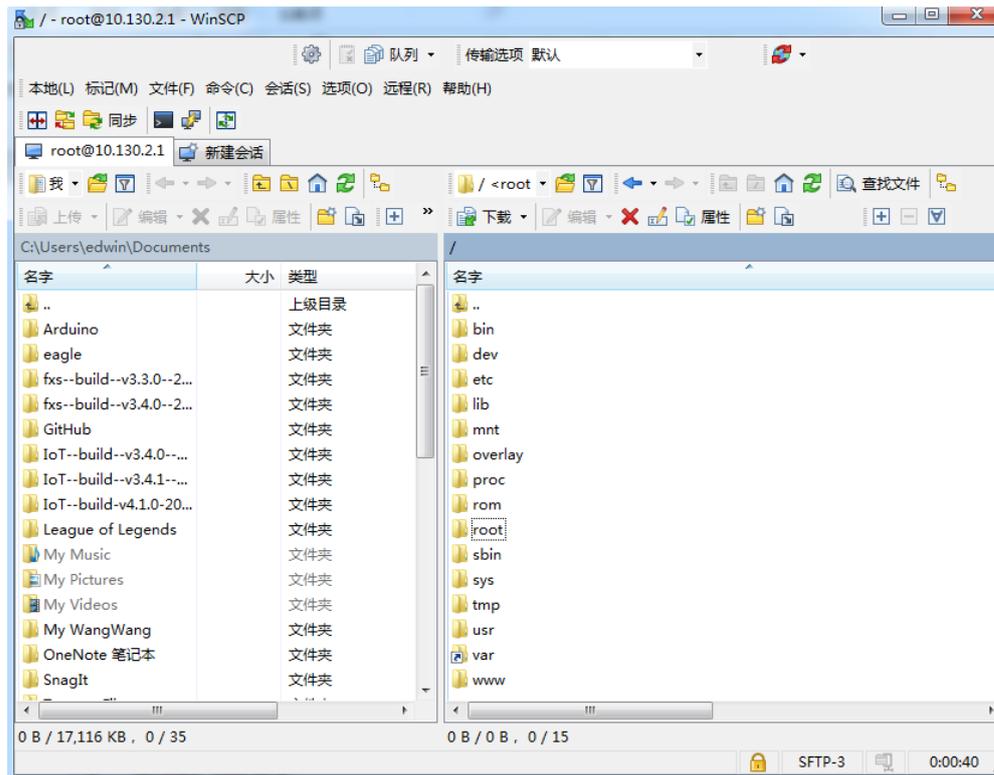
IP address: LG01 の IP アドレス  
 Port 番号: 22  
 User Name: **root**  
 Password: **dragino** (default)

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。



## 12 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&ドロップして転送したり、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：



## 13 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。

## 14 パッケージメインテナンスシステム

LG01 は、**OPKG パッケージメイ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：

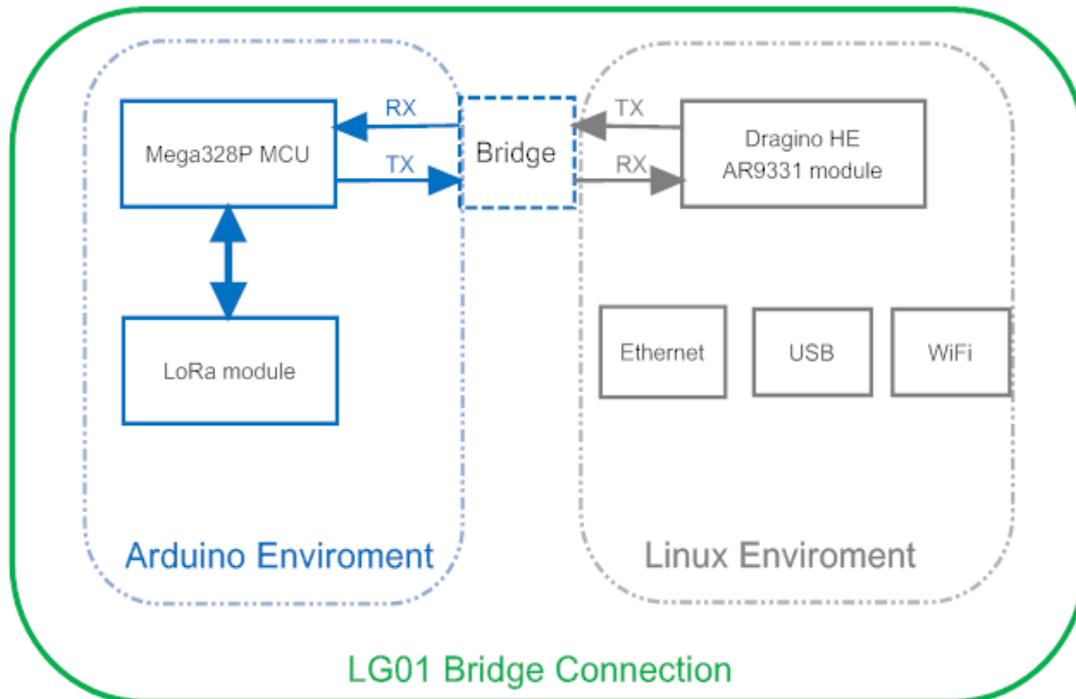
```
root@dragino-169d30:~# opkg update // 最新のパッケージリストを取得
root@dragino-169d30:~# opkg list //利用可能なパッケージを表示
root@dragino-169d30:~# opkg install mosquitto-client // MQTT クライアントをインストールし、要求したパッケージが自動インストールされます。
```

```
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.
```

## 15 ブリッジライブラリー

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ーの使用方法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。
- Arudino 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ー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。

## 16 アドバンスマネージメント

### 17 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)

## 18 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=Yun\\_Firmware\\_Change\\_Log](http://wiki.dragino.com/index.php?title=Yun_Firmware_Change_Log)

### dragino2-IoT--xxxxx-squashfs-sysupgrade.bin

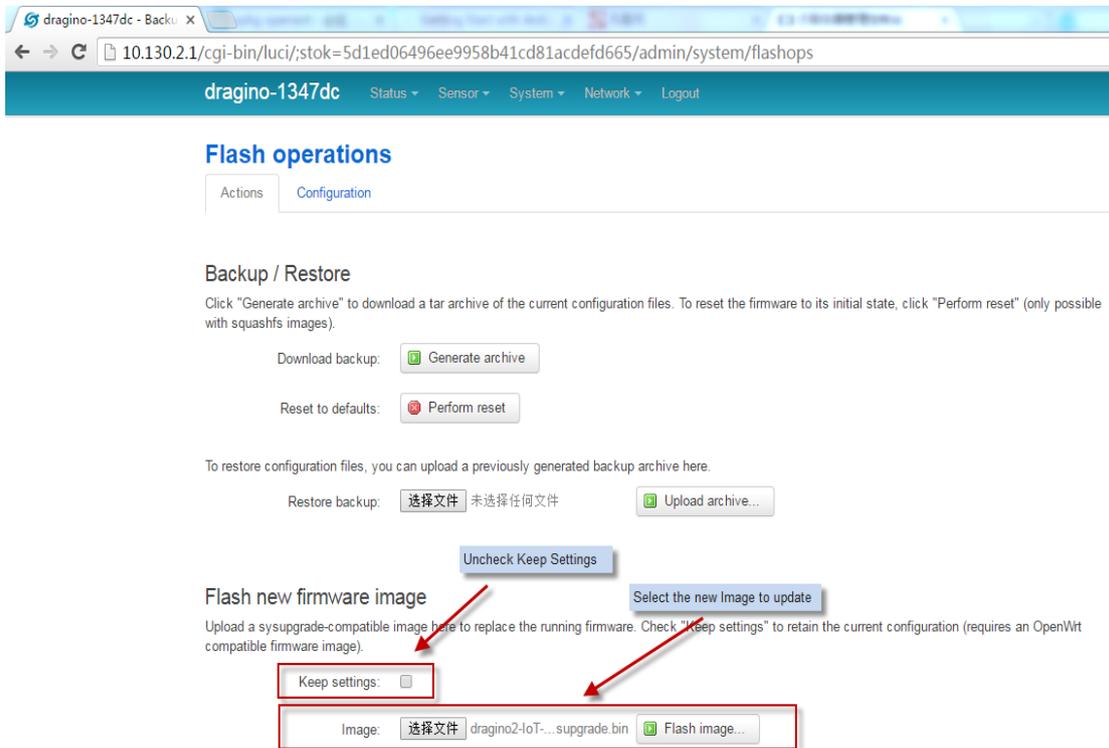
のファイル名は、アップグレードイメージです。アップグレードするにはいくつかの方法があります：

## 19 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で新規ファームウェアで自動ブートされます。



The screenshot shows the web interface for 'Flash operations'. It includes a navigation menu with 'Status', 'Sensor', 'System', 'Network', and 'Logout'. The main content area has two tabs: 'Actions' and 'Configuration'. Under the 'Backup / Restore' section, there are buttons for 'Generate archive' and 'Perform reset'. Below this, there is a section for 'Flash new firmware image' with a 'Keep settings' checkbox and an 'Image' field containing 'dragino2-1oT-...\_supgrade.bin'. A 'Flash image...' button is also present. Red arrows and boxes highlight the 'Uncheck Keep Settings' button and the 'Image' field.

## 20 Linux コンソールを経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。

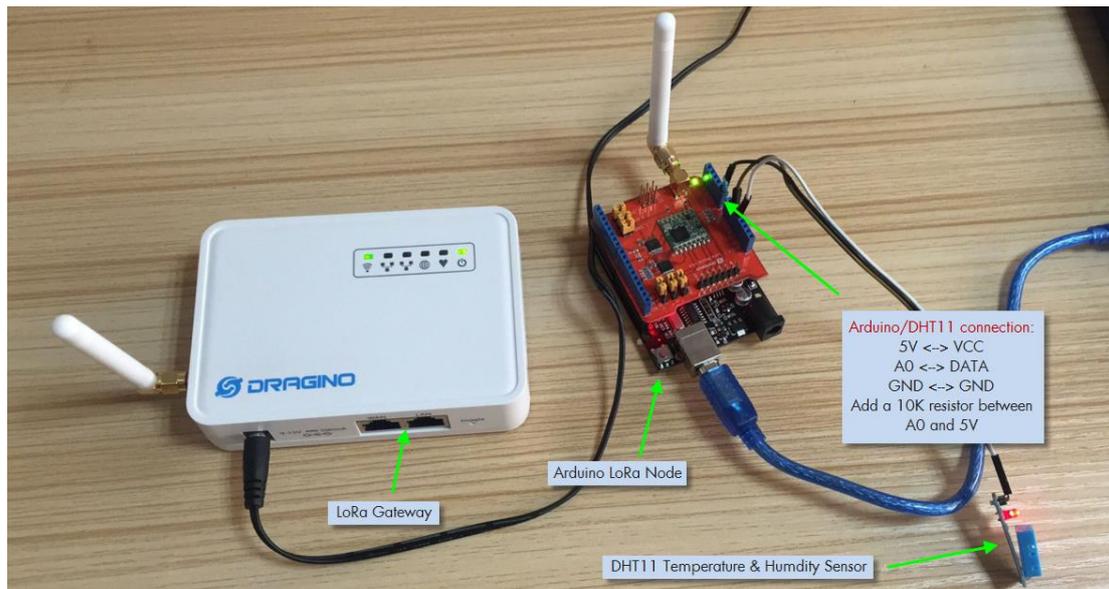
## 21 自動プロビジョニング

## 22 アドバンス事例

## 23 アップストリーム: LoRa クライアントノードからデータ取得し、IoT サーバーへ送信する

### 23.1.1 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。



### 23.1.2 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 ThingsSpeak 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. Channels have 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 location, tags, and units.
- **Tags:** Enter keywords that identify the channel. Separate tags with commas.
- **Latitude:** Specify the position of the sensor or thing that is sending data in degrees. For example, the latitude of the city of London is 51.5074.
- **Longitude:** Specify the position of the sensor or thing that is sending data in degrees. For example, the longitude of the city of London is 0.1278.
- **Elevation:** Specify the position of the sensor or thing that is sending data in meters.

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。

### 23.1.3 スケッチをアップロード

**Step 1:** 下記 URL より [ThingSpeak Library](#) 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** → **IoTSer** → **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。

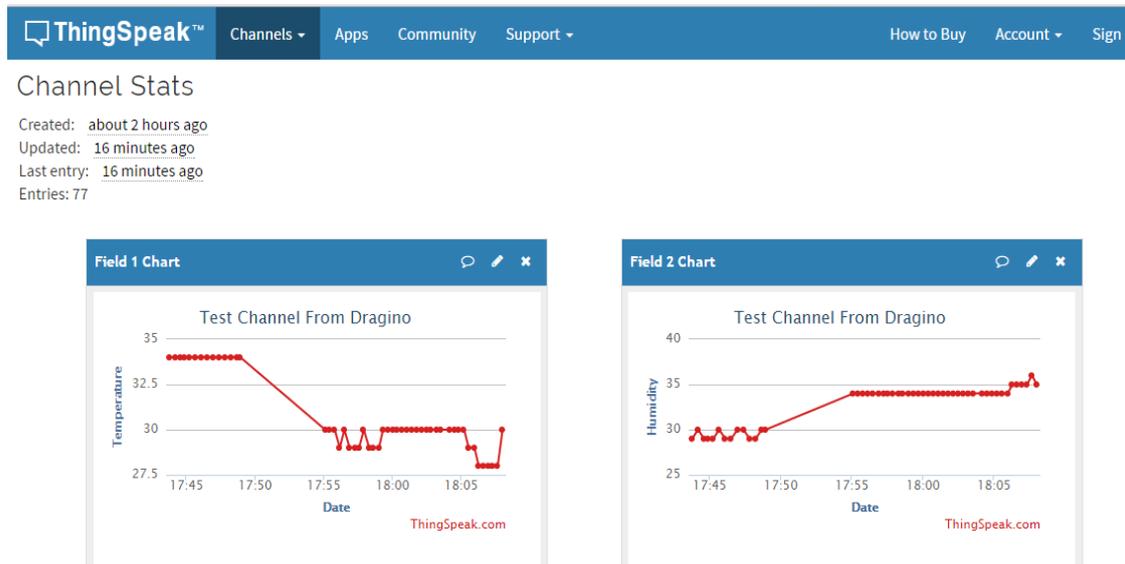
**Step 3:** 下記メニューよりサンプルを選択します:

➤ **IDE** → **File** → **Examples** → **Dragino** → **IoTSer** → **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。

### 23. 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。



## 24 ダウンストリーム: IoT サーバーからデータを取得し、LoRa クライアントノードに送信

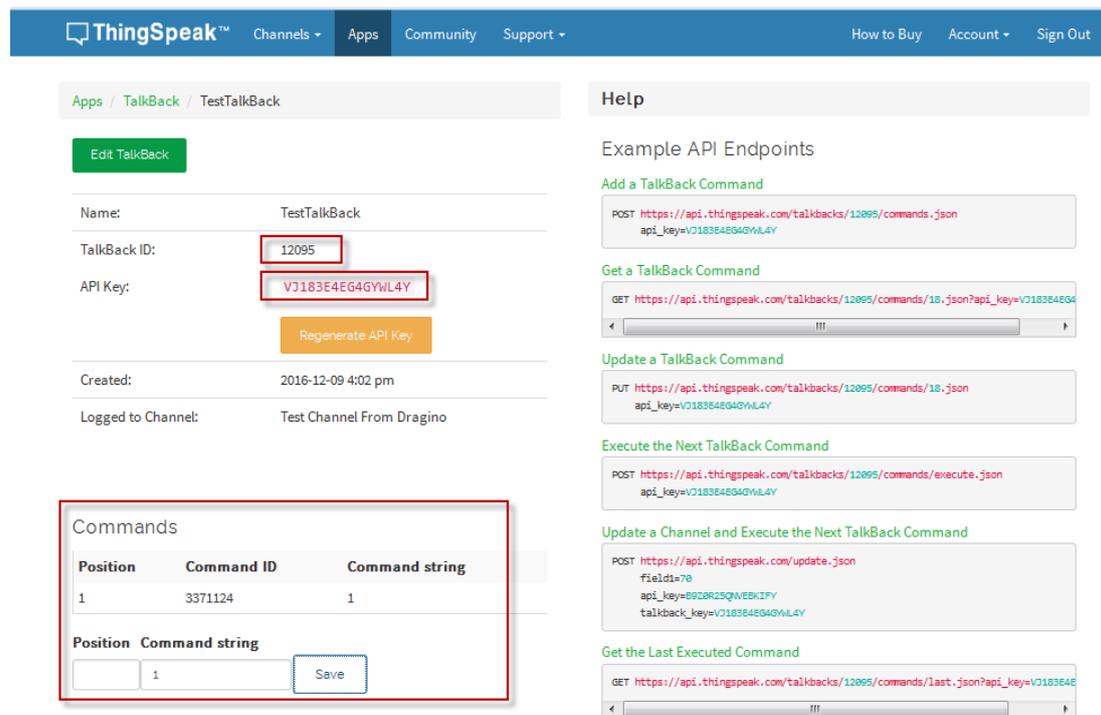
### 24. 1. 1 ハードウェア設定

下記写真のようにハードウェアを設定します。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。



### 24.1.2 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 shows the ThingSpeak web interface for configuring a TalkBack command. The main configuration area includes:

- Name:** TestTalkBack
- TalkBack ID:** 12095
- API Key:** VJ183E4E64GYWL4Y
- Created:** 2016-12-09 4:02 pm
- Logged to Channel:** Test Channel From Dragino

Below the configuration is a **Commands** table:

Position	Command ID	Command string
1	3371124	1

At the bottom of the table, there is a form to add a new command with a **Save** button.

On the right side, the **Help** section lists several API endpoints:

- Add a TalkBack Command:** POST `https://api.thingspeak.com/talkbacks/12095/commands.json` with `api_key=VJ183E4E64GYWL4Y`
- Get a TalkBack Command:** GET `https://api.thingspeak.com/talkbacks/12095/commands/18.json?api_key=VJ183E4E64GYWL4Y`
- Update a TalkBack Command:** PUT `https://api.thingspeak.com/talkbacks/12095/commands/18.json` with `api_key=VJ183E4E64GYWL4Y`
- Execute the Next TalkBack Command:** POST `https://api.thingspeak.com/talkbacks/12095/commands/execute.json` with `api_key=VJ183E4E64GYWL4Y`
- Update a Channel and Execute the Next TalkBack Command:** POST `https://api.thingspeak.com/update.json` with `field1=70`, `api_key=B92BR25QWBEKIFY`, and `talkback_key=VJ183E4E64GYWL4Y`
- Get the Last Executed Command:** GET `https://api.thingspeak.com/talkbacks/12095/commands/last.json?api_key=VJ183E4E64GYWL4Y`

### 24.1.3 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。

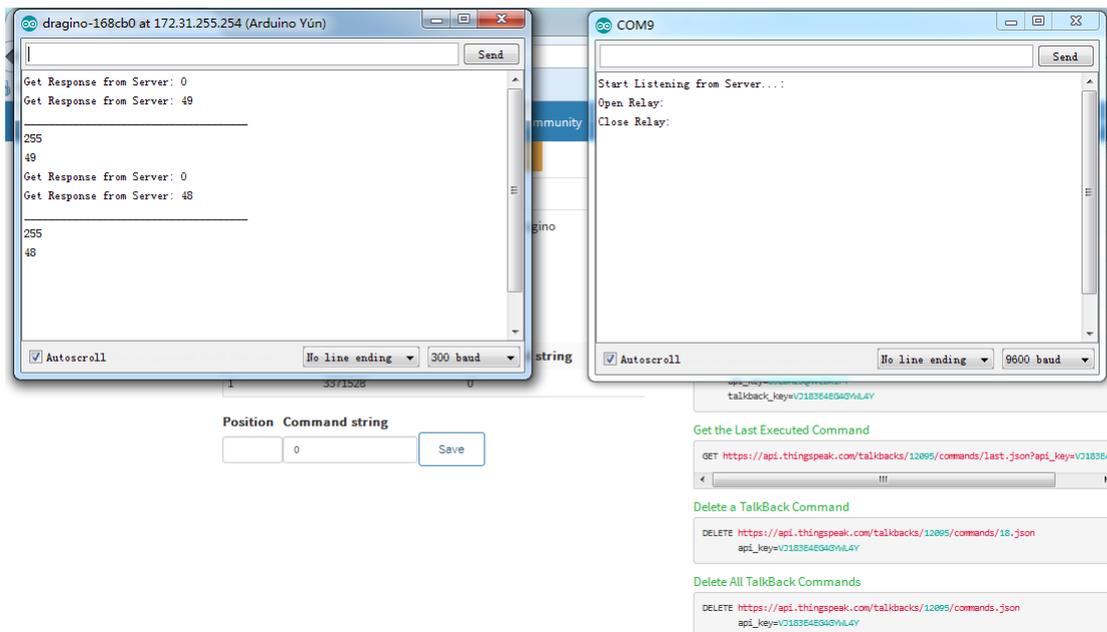
Step 2: 下記でサンプルを選択します：

- [IDE](#) → [File](#) → [Examples](#) → [Dragino](#) → [IoTServer](#) → [ThingSpeak](#) → [DownstreamServer](#)

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

### 24.1.4 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。



## 25 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](http://wiki.dragino.com/index.php?title=Examples_Catalog)

## 26 FAQ

### 27 何故、LoRa バージョンで 433/868/915/920 と分かれていますか？

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。

### 28 LG01 の LoRa バージョンでの周波数レンジはどのようになりますか？

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

### 29 LG01 のオリジナルファームウェアを作成することはできますか？ LG01 のソースコードはどこにありますか？

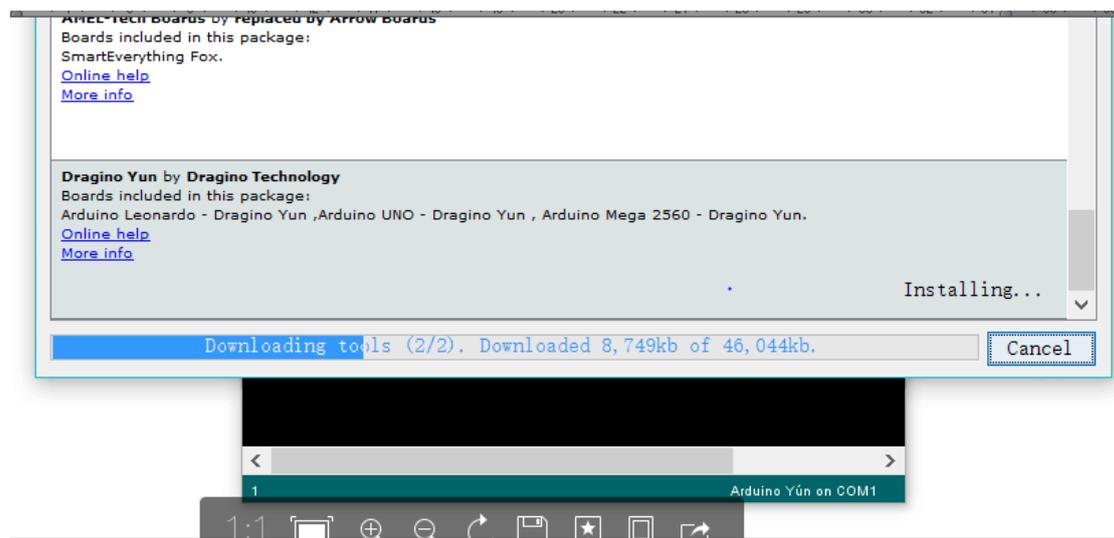
はい、独自のブランド化したり、カスタマイズアプリを追加したりして、LG01 オリジナルファームウェアを開発することはできます。LG01 のソースコードは下記で公開されています。そしてインストラクションに従い、コンパイルして再利用できます：

<https://github.com/dragino/openwrt-bb-1407>

### 30 トラブルシューティング:

#### 31 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:

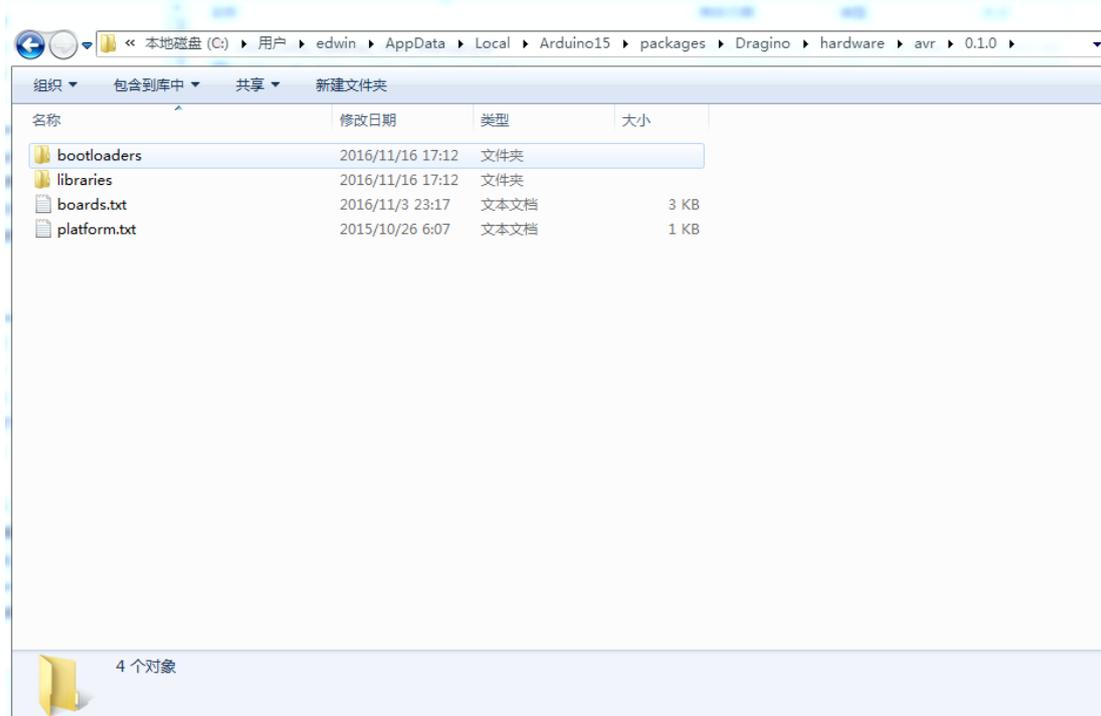
**STEP 1:** 下記 URL からプロファイルをダウンロードします。

<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ー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 ディレクトリー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。



### 32 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：

**STEP 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ーは、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。

### 33 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

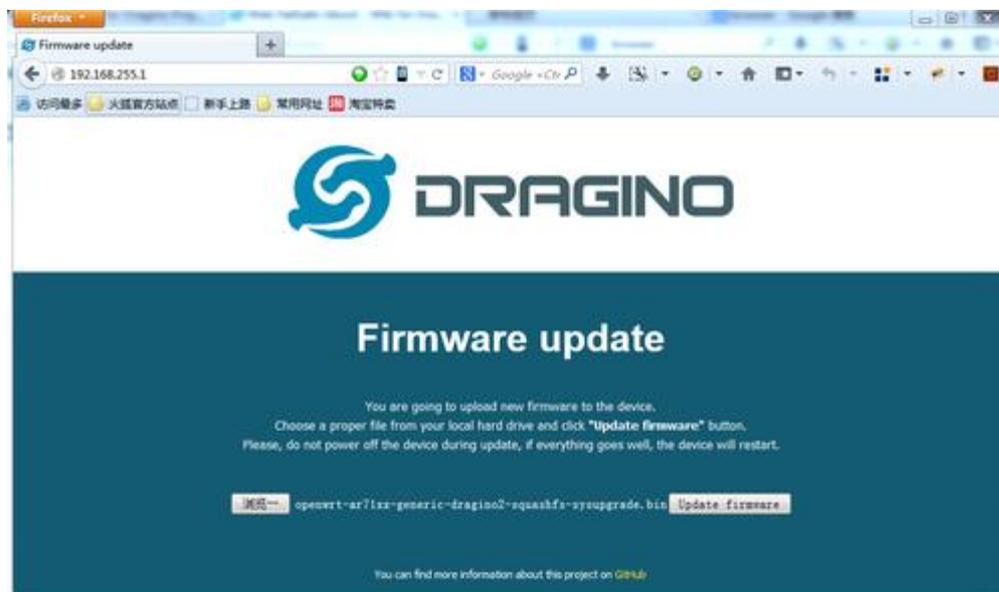
### 34 ファームウェアがクラッシュした場合、どうやって LG01 をどう回復できますか？

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。



### 35 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

### 36 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

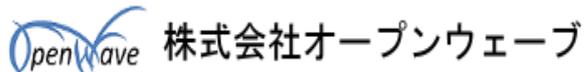
外寸と重量:

- ✓ 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

### 37. 参照ファイル

- ◇ LG01 オフィシャル wiki  
[http://wiki.dragino.com/index.php?title=Examples\\_Catalog](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-bb-1407>
- ◇ 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>

Dragino 国内総発売元



〒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](mailto: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](mailto:info@crijapan.jp)

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