

長距離×長時間 Sub-GHz 無線搭載
Lazurite ではじめる IoT 電子工作



国野 亘

内容

長距離×長時間 Sub-GHz 無線搭載 Lazurite ではじめる IoT 電子工作 ～ Sub-GHz ワイヤレス通信編 ～	3
1. IoT で注目される Sub-GHz.....	3
2. Sub-GHz 無線で長距離通信	4
3. Sub-GHz 無線対応 Lazurite の通信方式	4
4. Sub-GHz 実験準備①Lazurite 製品の構成.....	5
5. Sub-GHz 実験準備②ダウンロードとインストール.....	6
6. Sub-GHz 実験準備③プログラムを Lazurite へ書き込む	7
7. Sub-GHz 無線の通信実験を試みよう	8
8. Sub-GHz 送信機のプログラムを確認してみよう	9
9. 電池電圧や環境センサの測定値を送信してみよう	11
10. むすび (Sub-GHz ワイヤレス通信編)	11
長距離×長時間 Sub-GHz 無線搭載 Lazurite ではじめる IoT 電子工作 ～ クラウド接続編 ～.....	13
11. コイン電池で1年以上	13
12. クラウド接続準備①Lazurite の実験に必要な機器	14
13. クラウド接続準備②ドライバと Node-RED のインストール	15
14. Node-RED でセンサ値を受信する	18
15. クラウド接続準備③IoT 用クラウドサービス Ambient への登録	19
16. IoT 用クラウドサービス Ambient へ送信する	20
17. むすび (クラウド接続編)	23
参考文献	25
著者について	25
権利情報	25
履歴.....	25

長距離×長時間 Sub-GHz 無線搭載 Lazurite ではじめる IoT 電子工作 ～ Sub-GHz ワイヤレス通信編 ～

1. IoT で注目される Sub-GHz

パソコンなどの無線通信方式として Wi-Fi が普及していますが、IoT ではより長距離かつ省電力な通信方式が求められています。例えば、パソコンの Wi-Fi が受かりにくいときに、少し場所を変えると電波状態が良くなったり、Wi-Fi を使っていないときに比べてバッテリーの持ちが短かったりといった経験があると思います。固定して使う IoT 機器の場合、場所を変えたり、電池交換や充電を行ったりといったことがパソコンほど手軽では無いので、Wi-Fi の課題がパソコンよりも顕著に表れます。

これら 2つの課題を解決してくれるのが、Sub-GHz 無線を搭載した Lazurite です。Lazurite には、通信可能な距離が数 100m に及ぶ点や、乾電池やコイン型リチウム電池による長時間動作が可能な点に特徴があります。さらにネットワークの知識やハンダ付けの技能が不十分であってもワイヤレスセンサを簡単に製作することが出来ます。表 1-1 に筆者の主観による他の製品との比較表を示します。

本稿では Lazurite を用い、Sub-GHz の長距離通信や長時間駆動を実体験するための機器構成、開発環境のセットアップ方法、体感用のソフトウェアについて紹介します。



写真 1-1 Lazurite を使った Sub-GHz 実験の様子

数 100m の長距離通信やコイン型リチウム電池での動作が可能な Lazurite 920J から送信した情報を、Lazurite Sub-GHz で受信を行う実験を行ってみた。

表 1-1 Lazurite と類似した製品との比較 (筆者の主観による相対評価)

項目	Lazurite	Arduino	Raspberry Pi	ESP8266	XBee
参考モデル例	Lazurite Sub-GHz	Arduino UNO R3	Raspberry Pi 3	ESP-WROOM02-DEV	XBee ZB S2C
無線	ワイヤレス通信	○	—	○	○
	通信方式	802.15.4g/e	—	Wi-Fi	Wi-Fi
	長距離 Sub-GHz	○	—	—	—
	長時間通信機能	○	—	—	△
マイコン	プログラマブル	○	○	○	—
	マイコン性能	△	—	◎	○
	マイコン省電力	○	△	—	○
	IDE 使用言語	C 言語	Arduino 言語	Python 等	Arduino 言語 等
参考価格 (1 台)	7,506 円	3,240 円	4,800 円	1,280 円	2,500 円

2. Sub-GHz 無線で長距離通信

無線通信の距離を延ばすことは、IoT に限らず恒久の課題ですが、Sub-GHz 帯と呼ばれる 1GHz 未満の周波数を用いれば、比較的、容易に通信距離を延ばすことが出来ます。無線通信用の周波数の中でも、700~950MHz 付近の周波数帯は「プラチナバンド」とも呼ばれ、金（お金）よりも希少価値の高い資源の意味合いが感じられる特別な周波数帯です。これまで、この周波数帯の用途や利用方法、割り当てなどの見直しが幾度も行われてきました。

自由空間における電波の伝搬距離は、周波数の二乗に反比例するので、周波数が低いほど有利であると言われています。一方、アンテナの大きさも周波数に反比例するので、周波数が低いほど伝搬距離が伸びる現象の裏には、大きなアンテナを使うほど伝搬距離が伸びる現象が隠れています。

アンテナの大きさを筐体の大きさであると捉えると、電波を最も効率よく送受信できる筐体の大きさが見えてきます。例えば 900MHz の半波長は約 17cm で、この距離を隔てた空間に最も効率よく電位差を与え、また最も効率よく電位差を検知するには、1/4 波長となる 8cm から 3/8 波長となる 13cm 程度の大きさの筐体が必要です。この大きさは、ちょうどクレジットカードの大きさ（約 4 インチ）や、約 3.5 インチ~5 インチの全面液晶画面の端末などが相当し、人が手で持ち運ぶための最適な大きさであることがわかります。他にも、人間の身長に合わせて作られた建造物の中へ侵入しやすい点や、情報通信を行うのに必要な帯域幅が確保できる点、半導体やセラミック素子での損失が高い周波数に比べると少ない点など、好条件がそろっている周波数帯です。

Lazurite Sub-GHz や mini シリーズの Lazurite 920J には、920MHz~930MHz の Sub-GHz 帯の無線通信機能が搭載されており、Lazurite Sub-GHz の全長 10cm は、ちょうど効率よく電波の送受信が可能なサイズです。

ただし、貴重な周波数だけに、国によって Sub-GHz の周波数割り当てが異なります。Sub-GHz の IoT 機器は、多くの国で周波数割り当てが共通の Wi-Fi や、事業者が周波数調節可能な 4G 方式に比べて、周囲の他の機器へ及ぼす影響も大きくなります。したがって、必ず日本の電波法に基づく技術基準適合証明や工事設計認証を受けた製品を使ってください。不適切な機器を使った場合、実名の公表や、懲役、罰金が科せられる場合や、他の無線設備の利用者や事業者へ被害や損害を与える場合もあります。正規品の Lazurite を購入するか、技適マークを確認してください。



写真 2-1 Lazurite Sub-GHz の裏面アンテナ部

ローム製の通信モジュール BP3596A には、ヘリカル状の導電パターンが印刷された三菱材料製の誘電体チップ・アンテナ AM11DP-ST01 が実装されている。マイコンボードの全長と相まって、高効率な性能を発揮する。

写真 2-2 工事設計認証済のローム製 BP3596A

通信モジュール BP3596A のプリント基板には、工事設計認証済の無線設備であることを示す所定の表示が施されている。

3. Sub-GHz 無線対応 Lazurite の通信方式

Lazurite Sub-GHz や Lazurite 920J には、スマートメーターなどに搭載されている無線通信方式 IEEE

802.15.4g/4e が実装されており、統合開発環境 Lazurite IDE を使用することで、簡単に Sub-GHz の無線通信を行うことができます。

IEEE 802.15.4g/4e は、ZigBee で用いられていた IEEE 802.15.4 を元にスマートメーター用に仕様変更と拡張が行われた規格です。国内では 920MHz 帯を使った Wi-SUN 対応スマートメーターで使われており、電力メーターを順次交換してゆくことにより、1 家に 1 台以上の普及が確実に見込まれています。最低出荷数量がほぼ確定していることから、通信モジュールや IC の低価格化も予想することができ、IoT 分野での家庭内ネットワーク用や LPWA 用としても、期待されています。

執筆時点（2018 年 2 月）の Lazurite に搭載されている IEEE 802.15.4g/4e モジュールには、Wi-SUN 用のプロトコルスタックが実装されていないので、スマートメーターや Wi-SUN 機器との通信、Wi-SUN FAN による LPWA 通信などを行うことは出来ませんが、同じ周波数帯、同じ規格の無線通信方式の実力を体感することができます。

4. Sub-GHz 実験準備①Lazurite 製品の構成

Lazurite を使った Sub-GHz 実験に必要な機器を、写真 4-1 および表 4-1 に示します。Lazurite Sub-GHz は、一部の Arduino シールドの利用が可能な Sub-GHz 無線搭載マイコンボードです。本稿では、Arduino 用 LCD Keypad シールドを接続して使用します。Lazurite 920J は、Lazurite Sub-GHz と同等の機能を SD カードサイズに集約した小型マイコンボードです。

Lazurite 920J にファームウェアを書き込むには、別売りの Lazurite mini writer (Type B)を使って PC へ接続する必要があります。また、Lazurite Sub-GHz や Lazurite mini writer を PC へ接続するには、汎用の USB Micro-B ケーブル (Type-A→Micro-B) を使用します。

各 Lazurite 製品へ電源を供給するには、USB ケーブルで PC へ接続するか、USB 電源出力を搭載した AC アダプタやモバイル・バッテリーを使用します。AC アダプタやモバイル・バッテリーの中には、消費電流が少なくなると電源出力が自動的に切れるタイプのものもあり、長時間の実験を行いたい場合は、自動オフ機能の搭載されていないタイプか自動オフ機能を解除できるタイプのものを選んでください。

Lazurite 920J をコイン型リチウム電池で動かす場合は、スイッチサイエンス社が開発した Lazurite mini 用の各種基板が便利です。環境センサ基板 SSCI-031288 については後述します。



写真 4-1 Lazurite を使った Sub-GHz 実験に必要な機器
Sub-GHz 無線の送信機には、Lazurite mini シリーズの 920J (右上) を、受信機には Arduino 用 LCD Keypad シールド (左上) を接続した Lazurite Sub-GHz (左下) を用いた。ファームウェアを Lazurite mini シリーズへ書き込むには、Lazurite mini writer Type B (右上) が必要。

表 4-1 Lazurite を使った Sub-GHz 実験に必要な機器

製品名	内容
Lazurite Sub-GHz	Arduino シールドが使用可能な Sub-GHz 無線を搭載マイコンボード
Lazurite 920J	Lazurite Sub-GHz と同等機能を SD サイズに集約したマイコンボード
Lazurite mini writer	Lazurite 920J ヘファームウェアを書き込むための USB 接続ライター
LCD Keypad シールド	16 文字×2 行のテキスト文字表示が可能な Arduino シールド拡張ボード
USB Micro-B ケーブル	Lazurite をパソコンへ接続するための USB ケーブル (Type-A→Micro-B)
Lazurite Sub-GHz 用電源	DC9V 出力 AC アダプタ、USB 用 AC アダプタ、USB モバイル・バッテリー等
Lazurite 920J 用電源	USB 用 AC アダプタ、USB モバイル・バッテリー等

5. Sub-GHz 実験準備②ダウンロードとインストール

Lazurite IDE は、Lazurite Sub-GHz や Lazurite 920J 内のマイコン用のプログラム作成や、コンパイル、書き込みなどを行うための Windows 用ソフトウェア総合開発環境です。下記のウェブサイトから setup.exe をデスクトップなどへダウンロードし、ダブルクリックして図 6-2 のようにインストールを行います。また、USB シリアルドライバについても、同様にインストールしてください。Lazurite のウェブサイトにもインストール方法が記載されています。

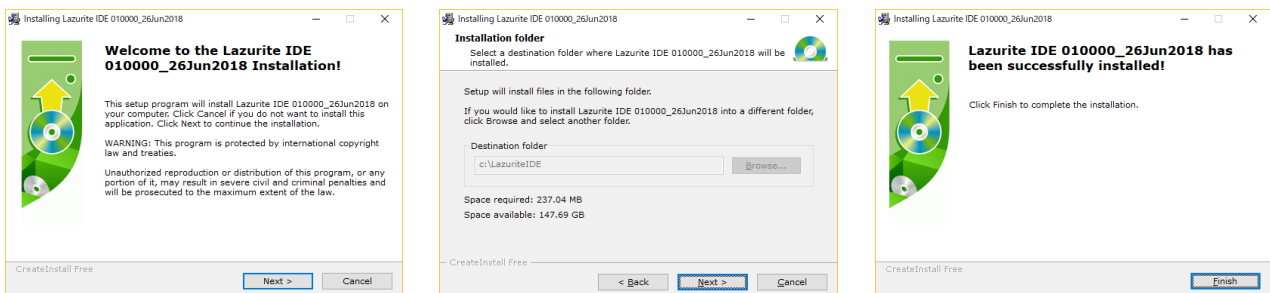
Lazurite IDE :
<http://www.lapis-semi.com/lazurite-jp/download>

USB シリアルドライバ :
<http://www.ftdichip.com/Drivers/VCP.htm>

インストール方法 :
<http://www.lapis-semi.com/lazurite-jp/contents/InstallGuide.html>

本稿で使用する筆者作成のプログラムが収録された ZIP ファイルは、下記からダウンロードすることが出来ます。ダウンロードした ZIP ファイルをダブルクリックし、ZIP ファイル内の lazurite-master フォルダをデスクトップやドキュメントフォルダなどへ保存してください。

サンプル・プログラム (筆者作成) :
<https://github.com/bokunimowakaru/lazurite/archive/master.zip>



1. setup.exe を起動し、Next をクリックする
2. インストール先を確認し、Next をクリックする
3. インストールが完了したら Finish をクリックする

図 5-1 Lazurite IDE のインストールの様子

6. Sub-GHz 実験準備③プログラムを Lazurite へ書き込む

ここでは、サンプル・プログラム practice_01_tx を Lazurite 920J へ書き込む方法について説明します。はじめに、写真 6-1 のように、Lazurite mini writer に付属するリボン・ケーブルを使って、Lazurite 920J と Lazurite mini writer を接続します。リボン・ケーブルの接続を行うときは、Lazurite mini writer の USB ケーブルを PC から外した状態にしておきます。リボン・ケーブルは、1 番ピン側の線が赤色で区別されている側になるように取り付けます。Lazurite 920J 基板の 1 番ピンには▲マークが、Lazurite mini writer 基板の 1 番ピンには「1」の数字が印刷されています。接続方向に誤りが無いかどうか、全てのピンがコネクタに接続されているかどうかを確認してから、USB ケーブルを PC へ接続してください。

Lazurite IDE を起動すると、図 6-1 のようなプログラムを選択する画面が開くので、lazurite-master フォルダ→1_practice フォルダ→practice01_tx フォルダ→practice_01_tx.c の順に開くと、図 6-2 のようにサンプル・プログラムが表示されます。

次に、プログラムを Lazurite 920J へ書き込むための設定を行います。このサンプル・プログラム practice_01_tx には、予め Lazurite 920J 用の設定ファイル(拡張子 ssf)が含まれているので、通信ポートの設定を行うだけで書き込むことができます。

通信ポートの設定を行うには、ツールメニュー内の[シリアル通信]の[通信ポート]から Lazurite を接続した COM ポート番号を選択します。COM ポート番号が分からないときは、同メニューの[通信ポート]に表示された全ての COM ポート番号を控えておき、Lazurite の USB ケーブルを抜き、再度、同メニューの[通信ポート]に表示される COM ポート番号を確認します。表示されなくなった COM ポート番号が Lazurite の COM ポート番号です。再び、USB ケーブルを接続し、Lazurite の COM ポート番号を選択してください。

プログラムを Lazurite 920J へ書き込むには、図 6-2 の右矢印のマークがあるアイコンをクリックします。書き込みが完了したら、Lazurite IDE 画面の下部に「Upload success.」と「Success to reset Lazurite」が表示されます。同じプログラムを Lazurite Sub-GHz へ書き込むことも可能です。その場合は、Lazurite IDE のツールメニュー内の[マイコンボード]から「LAZURITE Sub-GHz Rev2」を選択し、また、[オプション]から「LAZURITE Sub-GHz」を選択します。マイコンボード毎の設定値を表 6-1 に示します。



写真 6-1 Lazurite 920J へ Lazurite mini writer を接続する

Lazurite 920J と Lazurite mini writer をリボン・ケーブルで接続する。1 番ピン側の線が赤色で区別されている側を Lazurite 920J の▲マーク側と、Lazurite mini writer の「1」のマーク側にくるように接続する。写真では、それぞれのコネクタの右側 1 番ピンがある。

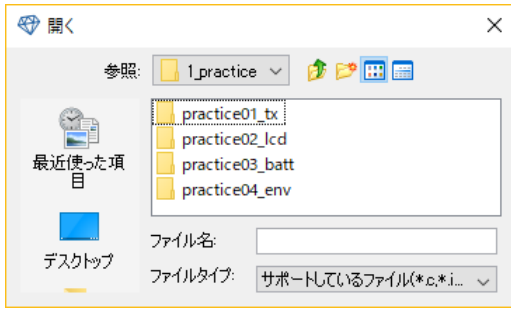


図 6-1 Lazurite IDE を起動し、プログラムを開く

Lazurite IDE を起動すると、プログラム選択ウィンドウが開くので、サンプル・プログラムをコピーした lazurite-master フォルダ内へ移動し、1_practice フォルダ→practice01_tx フォルダ→practice01_tx.c の順に開く。

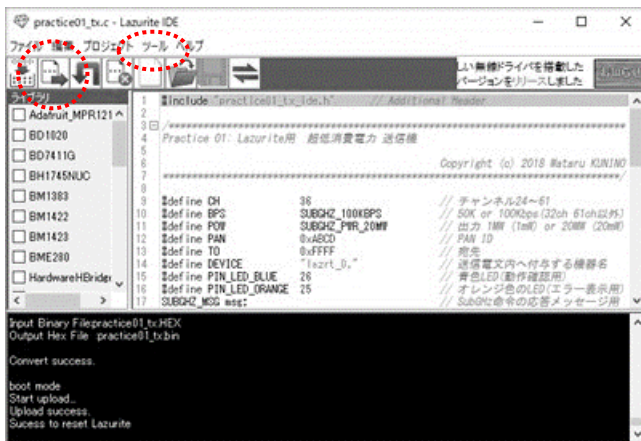


図 6-2 Lazurite IDE でプログラム practice_tx.c を開いたときの様子

プログラム practice_tx.c を開き、ツールメニュー内のマイコンボード、オプション、シリアル通信の設定を行ってから、右矢印のアイコンのあるボタンを押して、コンパイルと書き込みを実行する。

表 6-1 マイコンボード毎の Lazurite IDE のツールメニュー設定項目

メニュー名	PC へ接続したマイコンボード名と設定項目	
	Lazurite Sub-GHz	Lazurite 920J
マイコンボード	LAZURITE Sub-GHz Rev 2	LAZURITE mini
オプション	LAZURITE Sub-GHz	LAZURITE mini
シリアル通信	通信ポート	COM ポート番号を設定
	ボーレート	115200

7. Sub-GHz 無線の通信実験をしてみよう

それでは、実際に Sub-GHz 無線を使った通信実験を行ってみましょう。サンプル・プログラム practice_01_tx.c を Lazurite 920J へ、practice_02_lcd.c を Lazurite Sub-GHz へ書き込んでください。それぞれのマイコンボードへ割り当てられた COM ポート番号を、ツールメニューの[シリアル通信]の[通信ポート]で設定しておく必要があります。COM ポート番号以外のマイコンボード毎の設定項目を変更した場合は、変更した設定値を元の値に戻してから書き込んでください。

Lazurite Sub-GHz のボード上には、Arduino シールド用の拡張 IO 端子の信号電圧切り替えを行うためのスライドスイッチが実装されています。LCD Keypad シールドは 5V 仕様ですが、無線モジュールが 3.3V 仕様なので、スイッチを 3V 側に設定してください。スイッチの切り替えやシールド取り付け時は、Lazurite Sub-GHz に接続した USB ケーブルや AC アダプタを外した状態で接続し、シールドのピンがずれていないことを確認してから、電源を入れてください。

プログラム practice_01_tx.c を書き込んだ Lazurite 920J は、送信側です。起動後の経過秒数を約 3 秒間隔で送信します。受信機となる practice_02_lcd を書き込んだ Lazurite Sub-GHz の電源を入れると、LCD 上に「SubGHz Snif」の文字と、チャンネル番号 36ch、通信速度 100kbps、送信出力 20mW、PAN ID ABCD が表示されます。同じチャンネル番号かつ同じ通信速度の packets を受信すると、LCD の 1

行目に送信元のアドレス（末尾 2 バイト）、MAC フレームの種別コード（2 バイト）、送信シーケンス番号（1 バイト）、受信レベル RSSI（0～255）が表示されます。受信レベル RSSI の数字が大きいほど、受信強度が強いことを表し、RSSI 値×0.25dB から、およその受信感度マージンが計算できます。本機をモバイル・バッテリーで駆動すれば、家じゅうの様々な位置に持ち運び、宅内の受信レベルの様子を調べることが出来るでしょう。家の隅々まで電波が届くことが確認できたら、屋外へ持ち出しても良いでしょう。

LCD の 2 行目には受信したデータが表示されます。プログラム `practice_01_tx.c` からの送信であれば、データの先頭に「`lazrt_0`」が付与され、その後ろに、秒数が表示されます。なお、本機のプログラムではプロミスカスモードを使用し、PAN ID（ネットワーク内で共通の ID）が一致しなくても表示するようにしました。通常システムでは、プログラム `practice02_lcd.c` の 15 行目の `#define PROMISCUOUS` の `true` を `false` にして、プロミスカスモードを解除します。



写真 7-1 サンプル・プログラム `practice_02_lcd` を書き込んだ Lazurite Sub-GHz

`practice_02_lcd.c` を書き込んだ Lazurite Sub-GHz の電源を入れると、「SubGHz Snif」とともに、チャンネル番号 36ch、通信速度 100kbps、送信出力 20mW、PAN ID ABCD が表示される。

8. Sub-GHz 送信機のプログラムを確認してみよう

Lazurite 用のプログラムは C 言語で記述しますが、Arduino に似た書き方が出来ます。Sub-GHz 送信機用 `practice_01_tx` をプログラム 8-1 に、本プログラム内の主な手順①～⑩について以下に示します。

- ① プログラム先頭 1 行目の `#include` は、Lazurite IDE によって自動的に挿入されます。Lazurite IDE がライブラリを自動的に組み込むための仕組みなので、とくに気にする必要はありません。
- ② `#define` を使って、プログラム内で変更を行わない定数を定義します。
- ③ マイコン起動後に一度だけ実行する Arduino 互換の `setup` 関数を定義します。
- ④ GPIO モードを設定する Arduino 互換の `pinMode` 命令と、GPIO へ出力する Arduino 互換の `digitalWrite` 命令です。
- ⑤ `SubGHz.init` は Sub-GHz 無線の初期化を実行する命令です。
- ⑥ 継続的に繰り返し処理を行う Arduino 互換の `loop` 関数の定義です。
- ⑦ `Print.init` は文字配列型変数の初期化を行い、`Print.p` は文字列の、`Print.l` は数値の、`Print.ln` は改行の追記を文字配列型変数に対して行う Lazurite 独自の命令です。
- ⑧ `SubGHz.begin` は、通信を開始するための命令です。引数は、チャンネル番号、PAN ID、速度、送信出力です。それぞれの値は、手順②の `#define` で定義しています。
- ⑨ `SubGHz.send` は、データを送信する命令です。引数は、宛先、データ、データサイズと `NULL` です。受信完了時にコールバックする関数を 4 番目の引数で指定することも出来ます。

⑩ SubGHz.close は、通信を切断し、完了する命令です。

プログラム 8-1 Sub-GHz 送信機のプログラム

```

#include "practice01_tx_ide.h" // Additional Header

#define CH          36 // チャンネル 24~61
#define BPS         SUBGHZ_100KBPS // 50K or 100Kbps(32ch 61ch 以外)
#define POW         SUBGHZ_PWR_20MW // 出力 1MW (1mW) or 20MW (20mW)
#define PAN         0xABCD // PAN ID
#define TO          0xFFFF // 宛先
#define DEVICE      "lazrt_0," // 送信電文内へ付与する機器名
#define PIN_LED_BLUE 26 // 青色 LED(動作確認用)
#define PIN_LED_ORANGE 25 // オレンジ色の LED(エラー表示用)
SUBGHZ_MSG msg; // SubGHz 命令の応答メッセージ用

void setup() {
  pinMode(PIN_LED_BLUE, OUTPUT); // 青色 LED 用ポートを出力に設定
  pinMode(PIN_LED_ORANGE, OUTPUT); // オレンジ LED 用ポートを出力に
  digitalWrite(PIN_LED_BLUE, 1); // 青色 LED(動作確認用)を消灯
  digitalWrite(PIN_LED_ORANGE, 1); // オレンジ色 LED(エラー用)を消灯
  Serial.begin(115200); // 動作確認用シリアルレート設定
  if( (msg=SubGHz. init()) != 0) { // 初期化時にエラーが発生した時
    digitalWrite(PIN_LED_ORANGE, 0); // エラー表示用 LED を点灯する
    SubGHz. msgOut(msg); // エラーメッセージを表示する
    while(1) sleep(1000); // 永久に停止する(起動に失敗)
  }
  Serial.println("Practice 01 TX"); // 起動表示をシリアルモニタ出力
}

void loop() {
  char tx[17]; // 送信用の文字配列変数 tx を定義
  Print. init(tx, 17); // 変数 tx の初期化
  Print. p(DEVICE); // 変数 tx ヘデバイス名を代入
  Print. l(millis()/1000, DEC); // 変数 tx ヘ起動後経過時間を追加
  Print. ln(); // 変数 tx ヘ改行を追加
  Serial. print(tx); // 送信内容をシリアルモニタ出力

  digitalWrite(PIN_LED_BLUE, 0); // 動作確認用 LED を点灯
  SubGHz. begin(CH, PAN, BPS, POW); // 指定した条件で通信を開始する
  msg=SubGHz. send(PAN, TO, tx, sizeof(tx), NULL); // 変数 tx のデータを送信する
  if(msg) { // 応答値が SUBGHZ_OK 以外の時
    Serial. print("ERROR "); // エラー表示
    SubGHz. msgOut(msg); // 応答内容を表示
    digitalWrite(PIN_LED_ORANGE, 0); // エラー表示用 LED を点灯する
  }
  SubGHz. close(); // 通信を切断する
  digitalWrite(PIN_LED_BLUE, 1); // 動作確認用 LED を消灯
  sleep(3000); // 3 秒間、低消費電力状態で待機
}

```

9. 電池電圧や環境センサの測定値を送信してみよう

最後に、スイッチサイエンス社が開発した Lazurite mini 用環境センサ基板 SSCI-031288 を使い、電池電圧や温度、湿度、気圧を送信する実験を行ってみました。

写真 9-1 の環境センサ基板 SSCI-031288 は、ボッシュ製の温度・湿度・気圧センサ BME280 を搭載した、Lazurite 920J 用の拡張ボードです。コイン型リチウム電池 CR2032 で動作させることも可能です。なお、基板上の USB Micro コネクタは電源専用で、シリアル変換 IC は搭載されていません。

電池電圧を送信するサンプル・プログラムを `practice03_batt.c`、電池電圧に加えて温度・湿度・気圧を送信するサンプル・プログラムを `practice04_env.c` として収録しましたので、どちらかを Lazurite 920J へ書き込み、環境センサ基板 SSCI-031288 へ接続して実験してみてください。

受信機には前節と同様の `practice02_lcd.c` を書き込んだ Lazurite Sub-GHz を使用します。電池電圧を送信する `practice03_batt.c` からのデータを受信すると、LCD の 2 行目に `btttry_0` と、電池電圧 (mV) が表示されます。動作間隔は 3 秒、電圧の精度は 14 段階です。

ワイヤレス環境センサ `practice04_env.c` は、約 1 分ごとに、`envir_0`、温度 (°C)、湿度 (%)、気圧 (hPa)、電圧 (mV) を送信します。LCD の表示文字数を超えるので、気圧については上位 2 桁程度まで、電圧については表示されません。プログラム `practice02_lcd.c` の 77 行目を以下のように書き換え、`envir_0` の表示を省略すれば、温度、湿度、気圧、電圧の全てを表示することが出来ます。

プログラム 9-1 `practice02_lcd.c` の変更例

```
変更前:lcd_putstr(mac.payload);  
変更後:if(strlen(mac.payload)>8) lcd_putstr(mac.payload+8);
```



写真 9-1 温度・湿度・気圧センサ BME280 を搭載した Lazurite 920J 用拡張ボード

スイッチサイエンス社が開発した Lazurite mini 用環境センサ基板 SSCI-031288 には、ボッシュ製の環境センサ BME280 や、ローム製の照度・近接センサ RPR-0521RS、コイン型リチウム電池 CR2032 用のバッテリーホルダなどが実装されている。なお、Lazurite 920J 用コネクタは、ハンダ付けが必要。

10. むすび (Sub-GHz ワイヤレス通信編)

製作した機器を、家じゅうの様々な位置に設置してみたり、屋外に持ち出してみたりして、Sub-GHz の実力を確認してみると、従来の Wi-Fi との大差を実感することが出来ると思います。また、コイン型リチウム電池を使った送信機の電池電圧の推移を確認すると、超低消費電力の実力が分かるでしょう。

他の通信方式でこれらの性能を引き出すには、可能であったとしても専門的の知識が必要ですが、特別な配慮無しに簡単に引き出せることが、Lazurite 製品の最大の特徴です。

今回は、Sub-GHz 無線の区間の通信しか行いませんでしたが、機会があればインターネットとの連携についても紹介したいと思います。〈国野 亘〉



写真 10-1 受信機を屋外に持ち出し、Sub-GHz の RSSI レベルの測定を行ったときの様子

Lazurite Sub-GHz にモバイル・バッテリーを接続し、屋外に持ち出し、家屋から少し離れた場所で測定してみた。Sub-GHz の RSSI レベルは 71 で、受信感度点に対して約 18dB (距離換算で約 8 倍) の十分なマージンがあり、通信可能な距離の長さを実感することが出来た。

長距離×長時間 Sub-GHz 無線搭載 Lazurite ではじめる IoT 電子工作 ～ クラウド接続編 ～

11. コイン電池で1年以上

IoT 機器には様々な形態が考えられ、また一家に多くの機器が導入されると考えられています。人が持ち運ぶスマホやノートパソコンなどのモバイル機器と比較すると、普段は手の触れない場所に設置していることも多く、台数が増えてくると、電池交換や充電を行う煩わしさが増大します。このため、IoT センサには低消費電力動作が可能な通信機能を搭載したマイコンが必要です。Lazurite の場合も、コイン型リチウム電池 CR2032 で1年以上、単3アルカリ乾電池2本で10年以上の動作が、計算上、可能です。

本稿では Lazurite ゲートウェイをセットアップし、Lazurite から得られたセンサ情報を IoT 用クラウドサービス Ambient へ送信するシステムのセットアップ方法と、実験用ソフトウェアを紹介します。図 11-1 は、IoT センサ機器、IoT ゲートウェイ機器、クラウドサービスによるシステム構成例です。

IoT センサには、スイッチサイエンス社が開発した Lazurite mini 用の環境センサ基板 SSCI-031288 を用います。長距離通信実験で使用したソフトウェア practice04_env.c が書き込まれた Lazurite 920J を、写真 11-1 のように、センサ基板へ取り付けます。本ソフトウェアは「<https://goo.gl/WHf1g5>」からダウンロードすることが出来ます。Lazurite 920J への書き込み方法については前回の記事を参照してください。

IoT ゲートウェイのハードウェアは、写真 11-2 の Lazurite Pi Gateway を Raspberry Pi 3 へ装着して製作します。本ゲートウェイ用のソフトウェアには、IoT ゲートウェイをフロー図で設計することが可能な開発ツール Node-RED を用います。

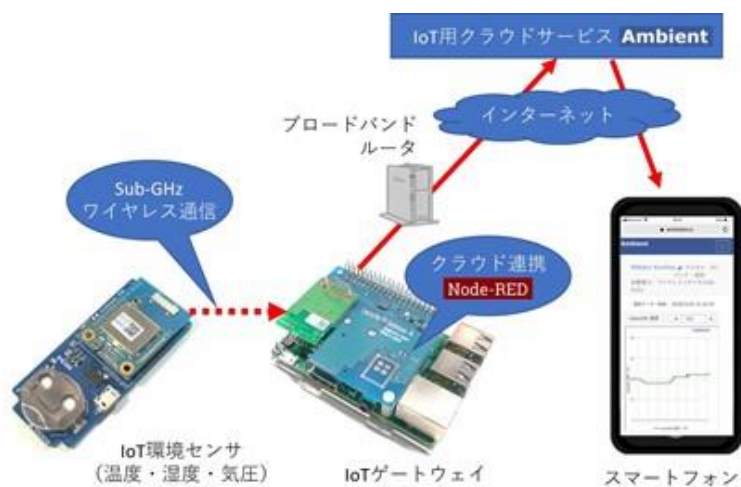


図 11-1 Lazurite を使ったクラウド接続システム構成図

IoT 環境センサ（左）は、温度、湿度、気圧情報を測定し、Sub-GHz ワイヤレス通信で IoT ゲートウェイ（中央）へ送信する。IoT ゲートウェイは、Sub-GHz ワイヤレス通信で受信したデータを選別し、IoT 用クラウドサービス Ambient へ転送する。



写真 11-1 IoT 環境センサ(温度、湿度、気圧)

Sub-GHz ワイヤレス通信機能を搭載した Lazurite 920J を、スイッチサイエンス製の Lazurite mini 用の環境センサ基板 SSCI-031288 に装着し、コイン型リチウム電池 CR2032 で動作させたときの様子。試算上、5 か月程度の動作が可能。

写真 11-2 Lazurite Pi Gateway

Raspberry Pi に装着することで、Sub-GHz ワイヤレス通信のゲートウェイを製作することが可能な無線モジュール。



12. クラウド接続準備①Lazurite の実験に必要な機器

Lazurite 製品によるクラウド接続実験に必要な機器を表 4-1 に示します。写真 11-1 の IoT センサ、写真 11-2 の Lazurite Pi Gateway のほか、写真 4-1 の Raspberry Pi 3、写真 12-2 の周辺機器、インターネットへの接続環境（無線 LAN または有線 LAN）などが必要です。

Lazurite 920J は、Sub-GHz ワイヤレス通信機能を搭載した SD カードサイズの小型マイコンボードです。内蔵のマイコンにファームウェアを書き込むには、PC と、別売りの Lazurite mini writer (Type B) と汎用の USB Micro-B ケーブル (Type-A→Micro-B) が必要です。

Lazurite Pi Gateway は、40 ピンコネクタを搭載した各種 Raspberry Pi シリーズに対応した拡張ボードです。試作や動作確認時は、開発に必要な周辺機器を手軽に接続することが可能な Raspberry Pi 2 または 3 に装着すると良いでしょう。長期間にわたる実験を行う場合は、低消費電力の Raspberry Pi Zero シリーズの特長を活かすのも良いでしょう。

Raspberry Pi を動作させるには、16GB の micro SD カードと、AC アダプタが必要です。一般的な利用方法に比べて、micro SD カードへのアクセスが増えるので、サンディスクや、東芝、パナソニックといった主力メーカーの製品や、MLC 方式のものを選ぶと良いでしょう。どのような micro SD カードであっても、バックアップと定期的なカード交換は必要ですが、交換頻度を抑えることができます。

表 12-1 Lazurite を使った Sub-GHz 実験に必要な機器

製品名	内容
Lazurite 920J	Sub-GHz 通信機能を内蔵した SD カードサイズのマイコンボード
Lazurite mini writer	Lazurite 920J へファームウェアを書き込むための USB 接続ライター
USB Micro-B ケーブル	Lazurite をパソコンへ接続するための USB ケーブル (Type-A→Micro-B)
Lazurite mini 用環境センサ基板	スイッチサイエンス社が開発した Lazurite 920J 専用の環境センサ基板
コイン型リチウム電池 CR2032	Lazurite mini 用環境センサ基板から Lazurite 920J への電源供給用
Lazurite Pi Gateway	Sub-GHz 通信モジュールを搭載した Raspberry Pi 用拡張ボード
Raspberry Pi 3	Linux システムを動作させることができる小型マイコンボード
micro SD カード 16GB	主力メーカー品や、MLC 方式などの高耐久品を推奨
Raspberry Pi 用 AC アダプタ	USB 5V 1.5A 以上を推奨 (一例: AD-B50P200)



写真 12-1 Linux システムの動作が可能な Raspberry Pi 3 有線 LAN や 4 個の USB 端子、モニタ出力用の HDMI 端子などを備え、Linux システムを動作させることが可能な小型のマイコンボード。ハードディスクの代わりに micro SD カードを使用する。



写真 12-2 Lazurite を使った Sub-GHz 実験に必要な周辺機器

ファームウェアを Lazurite mini シリーズへ書き込むための Lazurite mini writer Type B (左上) およびそのケーブル (左中)、Raspberry Pi用の micro SD カード (左下)、AC アダプタ (右上)、USB ケーブル (右下) が必要。

13. クラウド接続準備②ドライバと Node-RED のインストール

次に、Raspbian をインストールした Raspberry Pi へ、Lazurite Pi Gateway 用ドライバ・ソフトウェアと Node-RED をセットアップする方法について説明します。Lazurite Pi Gateway を Raspberry Pi の拡張用 GPIO 端子へ接続し、電源を投入してから、以下を実行してください。

- (1) Lazurite Pi Gateway は Raspberry Pi の SPI と I²C インターフェイスの両方を使用します。Raspberry Pi 上のメインメニューから [設定] を選択し、[Raspberry Pi の設定] (または Configuration) を起動し、[インターフェイス] (または Interfaces) の「SPI」と「I2C」を有効にし、[OK] をクリックしてください。再起動を促されたときは、再起動を実行してください。



図 13-1 Raspberry Pi の設定を開く

Raspberry Pi の画面右上のメインメニューから [設定] → [Raspberry Pi の設定] を開く

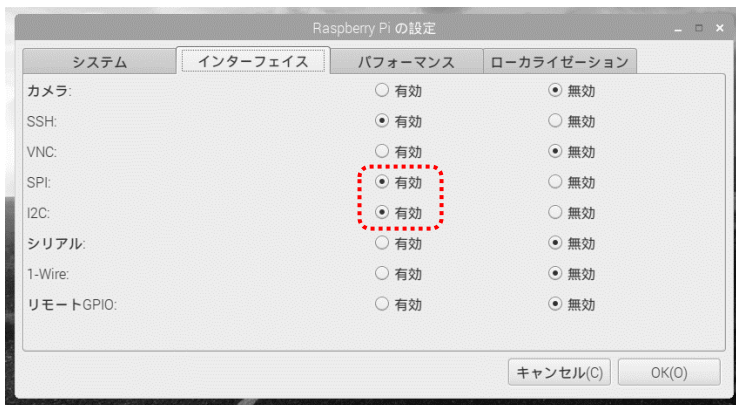


図 13-2 SPI と I2C を有効に設定する。

タブ [インターフェイス] 内の SPI と I2C を有効に設定する。

- (2) Lazurite Pi Gateway 用のシステム環境やドライバ・ソフトウェアなどのダウンロードとインストールを行います。まずはシステム環境のインストールです。Raspberry Pi 上のターミナル LXTerminal を開き、下記のコマンドを順に入力してください。



図 13-3 LXTerminal を開く。

```
cd↵
sudo apt-get update↵
git clone https://github.com/LAPIS-Lazurite/LazuriteInstaller↵
cd LazuriteInstaller↵
./setup.sh↵
```

setup.sh 実行中にエラーが出た場合は、ホームディレクトリへにダウンロードされた linux から始まる gz ファイルおよび、同じ Linux-から始まるディレクトリを削除し、再実行してください。それでも失敗する場合は、Raspbian を再インストールしてから再実行してください。

- (3) インストールしたシステム環境を有効化するために Raspberry Pi の再起動を行います。

```
sudo reboot↵
```

- (4) 再起動が終わったら、下記のコマンドで Lazurite 用ドライバとライブラリをインストールします。

```
cd LazuriteInstaller↵
./install.sh↵
```

- (5) 次に、以下のコマンドを入力し、Node-RED を起動します。起動時の画面例を図 13-4 に示します。メッセージ内に「Server now running」が表示されたら、Node-RED の準備完了です。

```
cd↵
node-red-start &↵
```

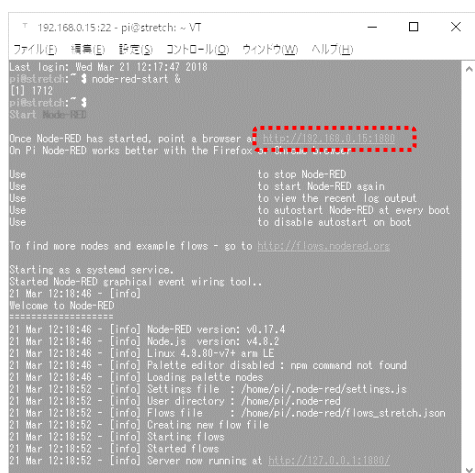


図 13-4 Node-RED を起動したときの画面例

Node-RED の画面を開くには、ターミナルソフトから Node-RED を起動し、メッセージ内に「Server now running」が表示されてから、インターネット・ブラウザで起動時の最初に表示される URL へアクセスする

- (6) Node-RED は、Raspberry Pi もしくは、PC 等のインターネットブラウザから操作します。Node-RED 起動時の最初に表示される「http://xxx.xxx.xxx.xxx:1880」をインターネットブラウザの URL 欄へ入力してください。PC は Raspberry Pi と同じ LAN に接続されている必要があります。

- (7) インターネットブラウザに Node-RED の画面が表示されたら、Lazurite 用のノードを Node-RED へ追加します。図 13-5 のように Node-RED の右上のメニューアイコン (3本の横線) をクリックし、[パレットの管理]を選択すると、図 13-6 のような「ユーザの設定」画面が開きます。続いて、[ノードを追加]タブを選択し、検索窓に「Lazurite」を入力し、「node-red-contrib-lazurite」が表示されたら[ノードを追加]をクリックしてください。追加には、1分くらいの時間を要することもあります。



図 13-5 Node-RED のメニューから [パレットの管理] を選択する



図 13-6 タブ [ノードを追加] を選択し、検索窓に「lazurite」を入力すると node-red-contrib-lazurite が見つかるので、[ノードを追加] ボタンをクリックする

ノードの追加に失敗する場合や、[パレットの管理] ボタンが表示されない場合や、Node-RED をアップデートしてください。Node-RED をアップデートする場合は、以下のコマンドを入力します。

```
cd
node-red-stop↵
update-nodejs-and-nodered↵
```

- (8) 同様の方法で IoT センサ用クラウドサービス Ambient のノードのインストールを行います。[パレットの管理] から [ノードを追加] タブを選択し、検索窓に「Ambient」と入力し、「node-red-contrib_ambient」の表示を確認してから [ノードを追加] をクリックしてください。



図 13-7 [パレットの管理]メニューでタブ[ノードを追加]を選択し、検索窓に「Ambient」を入力すると node-red-contrib-ambient が見つかるので、[ノードを追加]ボタンをクリックする

14. Node-RED でセンサ値を受信する

それでは、IoT センサから送られてきた情報を、製作した IoT ゲートウェイで受信し、Node-RED で実際に受信してみましょう。以下の手順と図 14-1 から図 14-3 のように、Node-RED 用の設定を行うことで、受信データを得ることが出来るようになります。

- (1) Node-RED の画面左側のパレットから、分類「Lazurite」内の Lazurite Rx ノードをマウスでドラッグし、中央のフロー画面へドロップしてください。Lazurite Rx ノードが配置されます。
- (2) 同じように、パレット内の分類「出力」の中の debug ノードをフロー画面へ配置します。配置後は「msg.payload」へ改名されます。
- (3) Lazurite Rx ノードの右端のコネクタから、ワイヤをドラッグし、msg.payload と名付けられた debug ノードへ接続してください。
- (4) Lazurite Rx ノードをダブルクリックすると、図 14-2 のようなノードのプロパティ画面が開くので、鉛筆アイコンをクリックし、図 14-3 のような Lazurite-channel ノードの編集画面を開きます。
- (5) Lazurite-channel ノードの編集画面の ch に「36」を、panid に「0xabcd」を、rate に「100kbps」を、Tx pwr に「20 mW」を入力し、[更新]ボタンをクリックしてください。
- (6) 画面右上の[デプロイ]ボタンをクリックするとフローの実行が開始されます。
- (7) 画面右側の[デバッグ]タブをクリックし、しばらく待つと、IoT センサからの受信データが表示されます。右側の情報画面を閉じてしまったときは、[Ctrl]+[スペース]キーを押すと表示されます。
- (8) フローの実行はブラウザを終了しても、続きます。フローの実行を停止させたいときや、実験が終わったときは、フロー図のタブをダブルクリックして、状態スイッチを[無効]に設定し、[デプロイ]を実行してください。

このように、Node-RED は、ワイヤで接続された各ノードのフロー図にしたがって、各種の IoT 機器から得られた情報を処理します。データを入力するための入力ノード、データ処理を行うための機能ノード、処理後の出力を行う出力ノードなどがあり、機能ノードには条件分岐や JavaScript による関数処理などがあります。左側のパレットの中から気になるノードを配置して、試してみても良いでしょう。

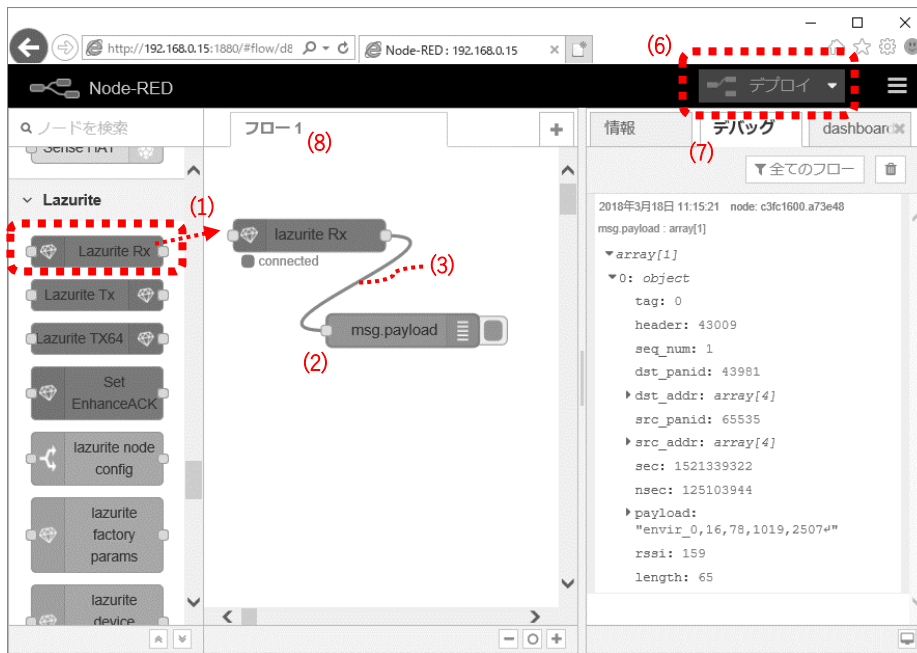


図 14-1 Node-RED のパレットからノード Lazurite Rx を配置し、Debug ノードへ接続し、受信してみたときの一例



図 14-2 Lazurite Rx ノードのプロパティ画面で鉛筆アイコンをクリックする



図 14-3 ch に「36」を、panid に「0xabcd」を、rate に「100kbps」を、Tx pwr に「20 mW」を入力し、[更新] ボタンをクリックする。

15. クラウド接続準備③IoT 用クラウドサービス Ambient への登録

IoT センサから得られたセンサ値を、IoT 用クラウドサービス Ambient へ送信することで、センサ値の蓄積やグラフ表示を行うことができます。Ambient は、アンビエントデータ株式会社が運営する IoT センサ用のクラウドサービスです。Ambient のサイト (<https://ambidata.io/>) にアクセスして、[ユーザー登録(無料)]のボタンからユーザー登録を行うことで、最大 64 データ (8 チャンネル×チャンネル当たり d1～d8 の 8 データ) のセンサ値を無料でアップロードすることができます。

メールアドレスと Ambient へのログイン用パスワードを登録すると、Ambient から案内メールが届きます。案内メールは迷惑メールと判断される場合があるので、届かなかった場合は、迷惑メールフォルダを確認してください。

メールには、<https://ambidata.io/usr/activate.html?t=> から始まる登録用 URL が記載されています。このリンクにアクセスすれば登録完了です。Ambient にログインすると図 15-1 のような My チャンネル画面が開くので、[チャンネルを作る]をクリックして、新しいチャンネルを作成してください。①チャンネル ID

は、センサ機器 1 台毎に割り当てられた番号です。②ライトキーは、センサ情報をクラウドへアップロードするときに使用するトークンです。これらは、後の手順で使用します。



図 15-1 Ambient にログインした時の My チャンネル画面

ログイン時に表示される My チャンネル画面で、①チャンネル ID と②ライトキーを確認する。

16. IoT 用クラウドサービス Ambient へ送信する

それでは、IoT 環境センサから得られたセンサ値を IoT 用クラウドサービス Ambient へ送信してみましょう。複数の Lazurite Rx が動作していると誤動作の原因になるので、Node-RED の画面上で、前回の Lazurite 用のフロー図のタブをダブルクリックし、状態スイッチを[無効]に設定して下さい。

使用するサンプル・フロー図データは、下記もしくは、ダウンロードした ZIP ファイル内の 3_node-red フォルダ内に含まれている node-red_example02.json です。

サンプル・フロー図データ（筆者作成）：

<https://goo.gl/tgzP6X>

- (1) インターネット・ブラウザまたはテキストエディタなどで JSON 形式のサンプル・フロー図データを開き、[Ctrl]+[A]で全選択を行い、[Ctrl]+[C]でクリップボードに保持してください。
- (2) Node-RED の右上のメニューアイコン（3本の横線）をクリックし、「読み込み」→「クリップボード」を選択し、[Ctrl]+[V]でサンプル・ノードを貼り付け、[読み込み]をクリックすると、フロー図のタブに[LzauriteEnvSensor]が追加されます。

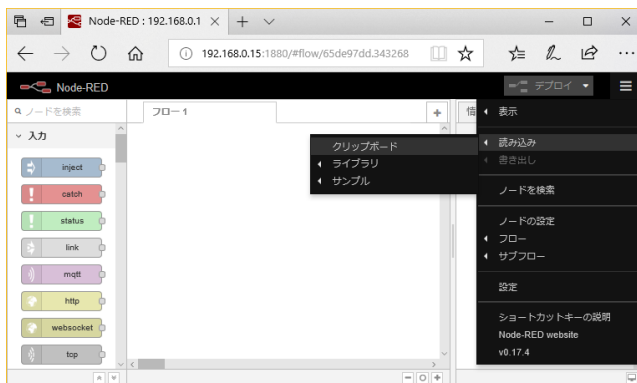


図 16-1 メニューアイコン（3本の横線）をクリックし、「読み込み」→「クリップボード」を選択する

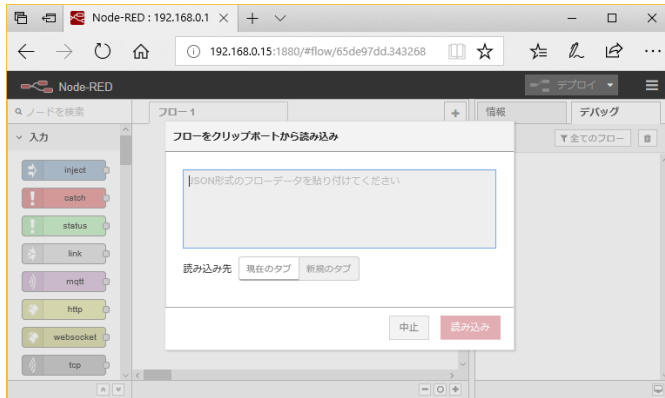


図 16-2 フローをクリップボードから読み込むウィンドウ内に、[Ctrl]+[V]でフロー図データを貼り付ける

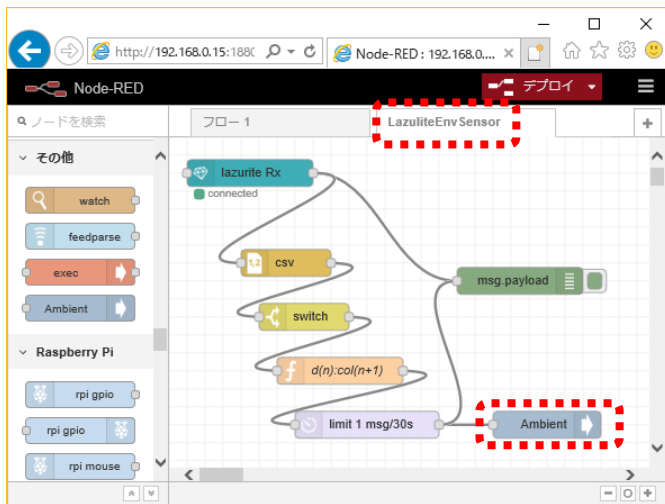


図 16-3 貼り付けられたフロー図データは、タブ LazuliteEnvSensor をクリックすると表示される

- (3) 貼り付けられたフロー図上の Ambient ノードをダブルクリックしてノードのプロパティ画面を開き、IoT 用クラウドサービス Ambient で取得したチャンネル ID とライトキーを、赤枠の ChannelId と WriteKey 欄へ入力し、[完了]をクリックしてください。

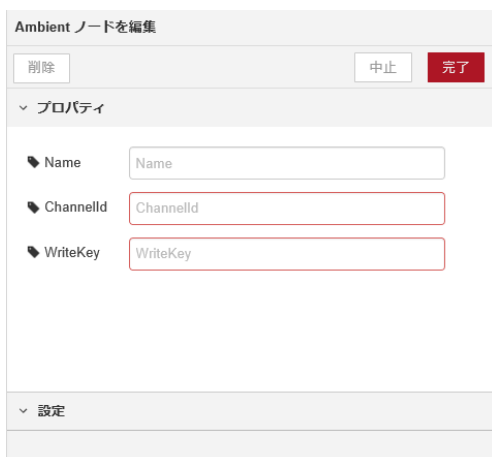


図 16-4 Ambient のプロパティ画面で取得したチャンネル ID を ChannelId 欄へ、ライトキーを WriteKey 欄へ入力し、[完了]ボタンをクリックする

- (4) 画面右上の[デプロイ]をクリックすると、フローが動作し始め、デバッグ画面に Lazulite からの受信データと Ambient への送信データが表示されます。

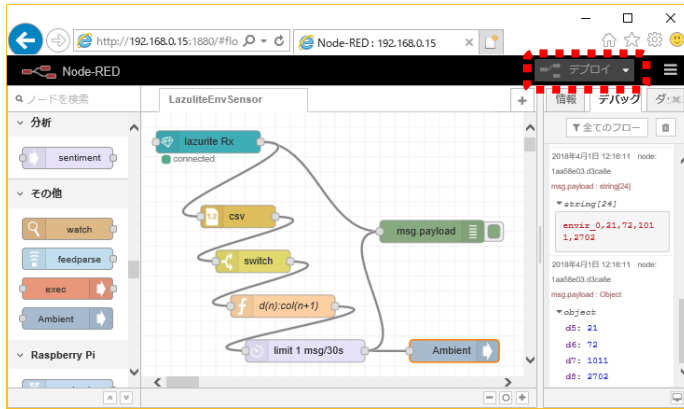


図 16-5 [デプロイ]をクリックすると、Lazurite から受信したセンサ値を Ambient へ送信しはじめる

(5) IoT 用クラウドサービス Ambient (<https://ambidata.io/>) へアクセスし、My チャンネルでチャンネル ID を選択すると、図 16-6 のような画面が表示されます。最新データ登録が「-」表示の場合は、データが正しく送信されていません。チャンネル ID、ライトキー（リードキーではない）が正しく設定されているかどうか、Raspberry Pi がインターネットへ接続されているかどうかを確認してください。



図 16-6 IoT用クラウドサービス Ambient のチャンネル表示の例

最新データ登録が「-」表示となっているときは、IoT センサのデータが送られてきていない。

(6) 最新データの日時が更新されているようであれば、図 16-7 のグラフ作成アイコンをクリックし、温度データ d1、湿度データ d2、気圧データ d3、電池電圧 d4 のそれぞれのグラフを画面へ追加します。

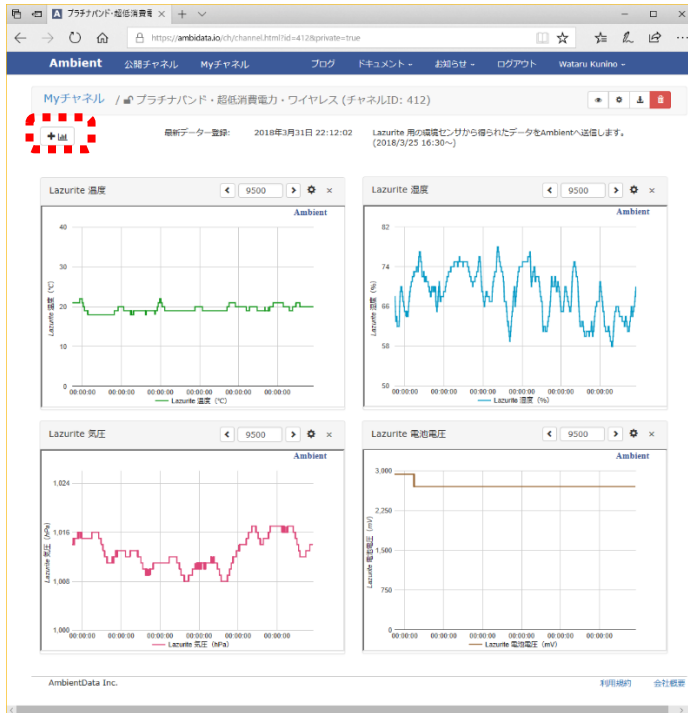


図 16-7 IoT 用クラウドサービス Ambient でセンサ値を表示したときの一例

温度データ d1 (左上)、湿度データ d2 (右上)、気圧データ d3 (左下)、電池電圧 d4 (右下) を表示した。

サンプル node-red_example02.json で設定した各ノードの動作内容は、以下の通りです。

- **Lazurite Rx ノード**
Lazurite Pi Gateway が受信したデータを出力する入力ノードです。ノードのプロパティ画面 (ノードをダブルクリックして表示) 中の format は出力形式です。「receive only latest packet」にチェックが入っており、受信したテキスト文字列を出力します。チェックを外すとオブジェクト形式で出力します。
- **csv ノード**
カンマ区切りのテキスト文字列をオブジェクト形式へ変換する機能ノードです。先頭から順にデバイス名が col1 に代入され、センサ値が col2 以降に代入されます。
- **switch ノード**
受信データに応じた分岐処理を行う機能ノードです。ここでは、デバイス名が envir_0 のときに次のノードへデータを渡します。プロパティに他の条件を追加することで、デバイス名に応じた異なる処理を行わせることも出来ます。
- **d(n):col(n+1)ノード (function ノード)**
受信したデータを Ambient 用の JSON 形式へ変換する function ノードです。変換方法はプロパティ内に JavaScript で記述されています。ここでは、col2~col5 の値を Ambient 用のデータ d1~d4 へ割り振ります。
- **Limit 1 msg/30 ノード (delay ノード)**
30 秒以内にデータ送信を行おうとしたときに、そのデータを破棄する delay ノードです。Ambient は 1 日あたりの 3000 サンプルを超えると、その日は保存できなくなるので、それを防止するためのノードです。
- **Ambient ノード**
入力されたデータを IoT 用クラウドサービス Ambient へ送信する出力ノードです。

17. むすび (クラウド接続編)

今回は、Lazurite の長時間駆動の性能を確認するための実験方法を紹介しました。消費電力からの試算では 5 か月程度の動作が可能ですが、その実証はこれからです。筆者はこれまで、XBee ZigBee 方式や Wi-Fi 方式などで乾電池駆動での長期間駆動の実証を行ってきましたが、試算時に考慮していなかった事象によって大きな影響が出ることもよくありました。例えば、Wi-Fi モジュール ESP-WROOM-02 を使

った 2015 年末の実験では、単 3 アルカリ乾電池 3 本の 30 分ごとの送信で 5 か月しか動作しませんでした。その後の改良により、1 年間の動作を達成できるようになりました。また、実証データ収集用のシステムの構築や、継続的な運用維持に多くの手間と労力を要しますが、図 17-1 のように Ambient を利用することで、手軽に実験が行えるようになりました。

Lazurite の省電力性能は明らかですが、実際にボタン電池 CR2032 による 1 分ごとの送信を行い、試算との差異を確認してみると、新たな課題を発見できるかもしれません。〈国野 亘〉



図 17-1 IoT 用クラウドサービス Ambient で Wi-Fi モジュール ESP-WROOM-02 を 1 年間動作させたときの様子

トランジスタ技術 2016 年 9 月号 P.90 図 4 の回路「Wi-Fi 温湿度センサ」に、大容量コンデンサを追加し、単 3 アルカリ電池×3 本で、12 か月以上の動作が可能であることを確認した。

参考文献

本書を作成するにあたり，下記の文献を参考にしました．

- <http://www.lapis-semi.com/lazurite-jp> [※当該サイトは 2022 年に閉鎖された]
2015 年 10 月～2020 年 8 月（ラピスセミコンダクタ株式会社）
- 920MHz 帯特定小電力無線モジュール BP3596A Datasheet
2015 年 7 月（ローム株式会社）
- ML620Q503H/Q504H ユーザーズマニュアル
2015 年 8 月 31 日（ラピスセミコンダクタ株式会社）

著者について

国野 亘（くにの・わたる）

ボクにもわかる電子工作 管理人

<https://bokunimo.net/>

関西生まれ。言葉の異なる関東や欧米などさまざまな地域で暮らすも，近年は住みよい関西圏に生息し続けている哺乳類・サル目・ヒト属・関西人。おもにホビー向けのワイヤレス応用システムの研究開発を行い，その成果を書籍やウェブサイトで公開している。

権利情報

本書の著作権は筆者（国野 亘）に帰属します。

著者（国野 亘）に無断で本書の複製や改変，販売，配布などを行うことを禁止いたします（適切な引用や，筆者からの承諾を得ていた場合を除く）。

違反した場合，当方や本書の関係者が被った損害額に加え，調査・相談・訴訟等に要した全費用と将来に得られたと推測できる被害額等を弁償していただく場合があります。

本書の掲載事項によって生じたいかなる損害についても，著作者は一切の責任を負いません。すべて自己責任にてご利用ください。

本文中では，製品名，会社名の商標表示を省略しています。

Copyright (C) 2018-2023 国野 亘 (<https://bokunimo.net/>)

履歴

2018 年 2 月	調査・執筆期間
2018 年 3 月 5 日	執筆完了(Sub-GHz ワイヤレス通信編)
2018 年 3 月	執筆期間(クラウド接続編)
2018 年 4 月 1 日	執筆完了(クラウド接続編)
2023 年 5 月 6 日	Web 公開