本マニュアルは、３G IoT Module のご利用に当たっての説明資料です。
一部の機能詳細においては、既に販売していますIEM版３Gシールドの資料等をご参照ください。

# ３GIM（3G IoT Module）利用マニュアル

株式会社 タブレイン

３GIM-V1.1R05

※本マニュアルは、３GIM V1.0および ３G シールドでもご利用頂けます。

20151110

もくじ




# 第1章
# 3GIM（3G IoT Module）の概要

# 1．はじめに

- ３GIMは、M2MやIoTシステム開発での試作やプロトタイプから、量産化に向けた３G通信モジュールとなります。
- 今後の新しいモノづくりにおいて、広域ワイヤレス（３G通信）を誰もが、高度な技術を、簡単に、短時間で、利用し、応用できるようにしたものが３GIMとなります。
- 3GIM（３G IoT Module）は、様々なマイコンを使って、簡単にインターネット接続することができるSDカードサイズの超小型3G通信モジュールです。
- 従来の3GシールドはArduinoでの利用をターゲットとしていましたが、3GIMでは様々なマイコン(mbed,GR-Sakura,PIC,Raspberry Pi等)からUART経由またはUSB経由で簡単に利用することができます。
- 「**IEM版３GシールドV2.0**」の関連ドキュメント類も参考にしながら開発を進められることをお薦め致します。
- 本製品の技術サポートおよび今後のマニュアル更新につきましては、以下のWikiページをご覧ください。

  http://a3gs.wiki.fc2.com/

- 本製品購入時には、①３GIM本体、② ３G（GPS併用）アンテナ＋コネクタ・ケーブル、③ 本説明書含む資料のダウンロードサイトURL （PW）が提供されます。
- Wikiページでご紹介していますように、mbed、RaspberryPi、Edison、Galileoなどでも３GIMが利用できています。

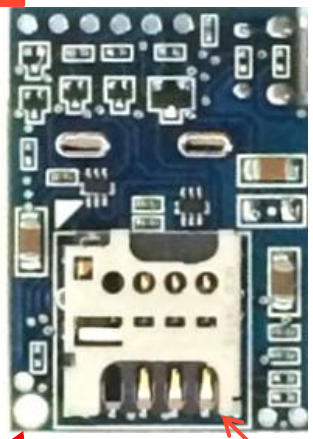
# 2．3GIMの外観

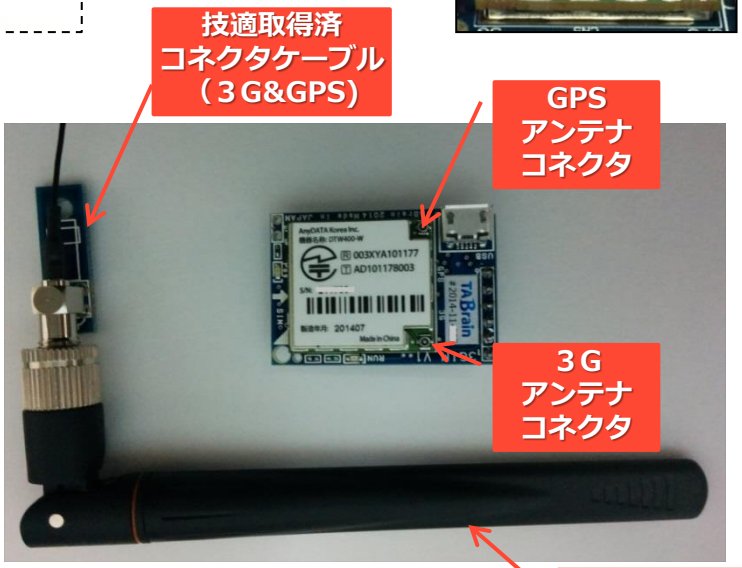
マイクロSIMの挿入方向

※マイクロSIMにテープを付けて出し入れできるような工夫があると便利です

※コネクタケーブルをコネクタの上から押し込みます

## ■3GIMの外観写真およびその説明


- 表側には、IEM、マイクロUSBコネクタ、電源LED、シリアル番号シール等が配置されています。またシール上の脇には、**6ピンコネクタ**があります。
- 裏側には、**マイクロSIM**(ミニSIM)**ソケット**があります。（SIMカード挿し込む時、裏表・上下を間違わないように）
- アンテナ・コネクタは、専用の３GおよびGPSのアンテナ・ケーブルをご利用ください。
  （アンテナとケーブル部分は、特に扱いに注意が必要となります）

※アンテナ＋コネクタ・ケーブルは、ご購入時に３G（GPS併用）用１組が提供されます。GPSをご利用の場合には、別途ご購入する必要があります。

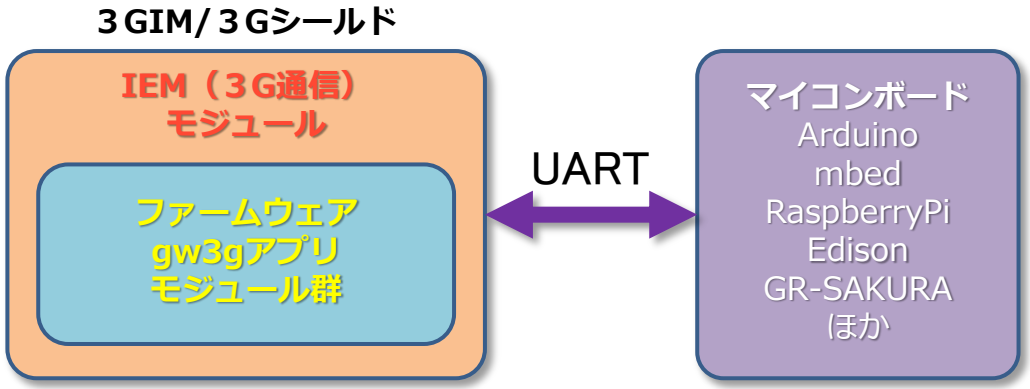
# 3．3GIMの機能概要

- **3GIMには、下記の機能（3GシールドV2.0と同じ）が提供されます。**（**gw3gアプリ**の機能）
  1. 3Gを介したインターネット接続（TCP/IPおよびHTTP、HTTPS※1）
  2. 3Gネットワークからの時刻取得
  3. GPSを使った位置情報の取得
  4. SMSの送受信（※2）
  5. 小容量のストレージ機能（通信モジュール上のメモリを利用）
  6. その他機能（電波強度の取得、ボーレートの変更、APN※3の切り替え等）

※1：対応できるSSL証明書には一部制約があります。すべてのサーバに対してHTTPS通信ができることを保証する訳ではありません。

※2：SMSの送受信はSMS不達の時の再受信動作に問題があり、受信機能の利用はお勧めしません。送信機能は特に問題なく利用いただけます。

※3：利用可能なSIMカードについては、「利用上の留意点」をご覧ください。



# 4．3GIMの仕様概要

| 項　目 | 仕　様 | 補　足 |
|---|---|---|
| 外形寸法 | 幅25mm * 奥行35mm * 高さ8mm | **取付穴**はø2.6(1ヶ所) |
| 電源電圧 | **電源コネクタ部**　3.6～4.2V<br>（**注意**：3.3Vおよび5Vは使用不可） | 安定したDC電源または3.7Vリチウムポリマ電池**※1**を推奨 |
| 消費電流 | 50～800mA | 利用状況や電波状態に依存 |
| 通信規格・対応周波数 | 3Gシールドと同じ | |
| マイコンとのインタフェース | UARTを介したコマンド・レスポンス方式またはUSBモデム | 仕様書は別途公開予定(ただし、USBモデムは規格のみ公開**※2**) |
| 使用アンテナ | 同梱するポール型アンテナ | 取付用コネクタおよび基板を標準で添付 |
| ロジック電圧 | 任意のロジック電圧で利用可能(3GIMにIO電圧を供給) | |
| UART | 9600～57600bps/8データビット/パリティなし/1ストップビット | ボーレートは変更可 |

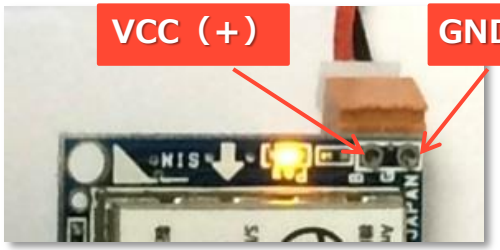
**※1**：USBコネクタ経由で充電は可能ですが、製品機能としてはサポート外・保証外とさせていただきます。
**※2**：USBモデムとしてのご利用に関しては、技術サポートは行っていません。

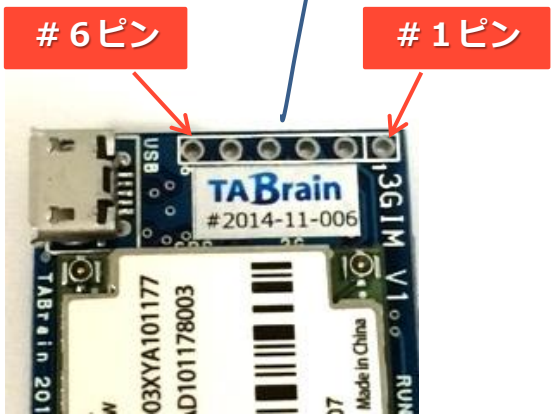
# 5．３GIMのピンコネクタ配置

| ピン番号 | 名称 | 機能など |
|---|---|---|
| ＃1 | PWR_ON | 電源のON/OFF制御(開放または0:LOWでON、1:HIGHでOFF) |
| ＃2 | RX | UARTインタフェース(RX)：相手方のTxに接続 |
| ＃3 | TX | UARTインタフェース(TX)：相手方のRxに接続 |
| ＃4 | IOREF | ロジック電圧(3.3V 又は 5V) |
| ＃5 | VCC（+） | **電源電圧(3.6～4.2V)** 又は 開放し「**外部電源**」から供給 |
| ＃6 | GND（−） | グラウンド |

【補足説明】

- ＃1ピンは「1」とシルク印刷されている側のピンです。
- ＃1ピンの「HIGH」は、ロジック電圧(IOREF)とします。
- 「VCC(+)」ピンから電源を供給する場合は、 PWR_ONの状態によらず、常にONとなります




注意：VCCを間違わないように
推奨：3.7Vリチウムイオン電池推奨

※ **VCC**または#5から電源供給必要
電源電圧（3.6~4.2V：600mA以上）
＜#1で電源ON/OFF制御＞

# 6．ご利用上の留意点

1. docomoのFOMA回線を利用します。そのため、docomoあるいはそのネットワークを利用するMVNOが提供するマイクロSIMが利用できます（ただし、これらの条件を満たす全てのSIMカードでの利用を保証する訳ではありません。利用できるSIMカードはWikiサイトのSIMカード情報ページを参照ください）

   http://a3gs.wiki.fc2.com/

   ※提供されている以外のNTTドコモ製のSIMカードご利用されたい場合には、別途info@tabrain.jpへご連絡ださい。

2. 日本国内での利用をお願いします。海外では、各国の法律により現状ではご利用いただけません。詳細はタブレインまでご相談ください。

3. USBモデムとして利用する場合でも、電源供給（3.6~4.2V）は必要です。

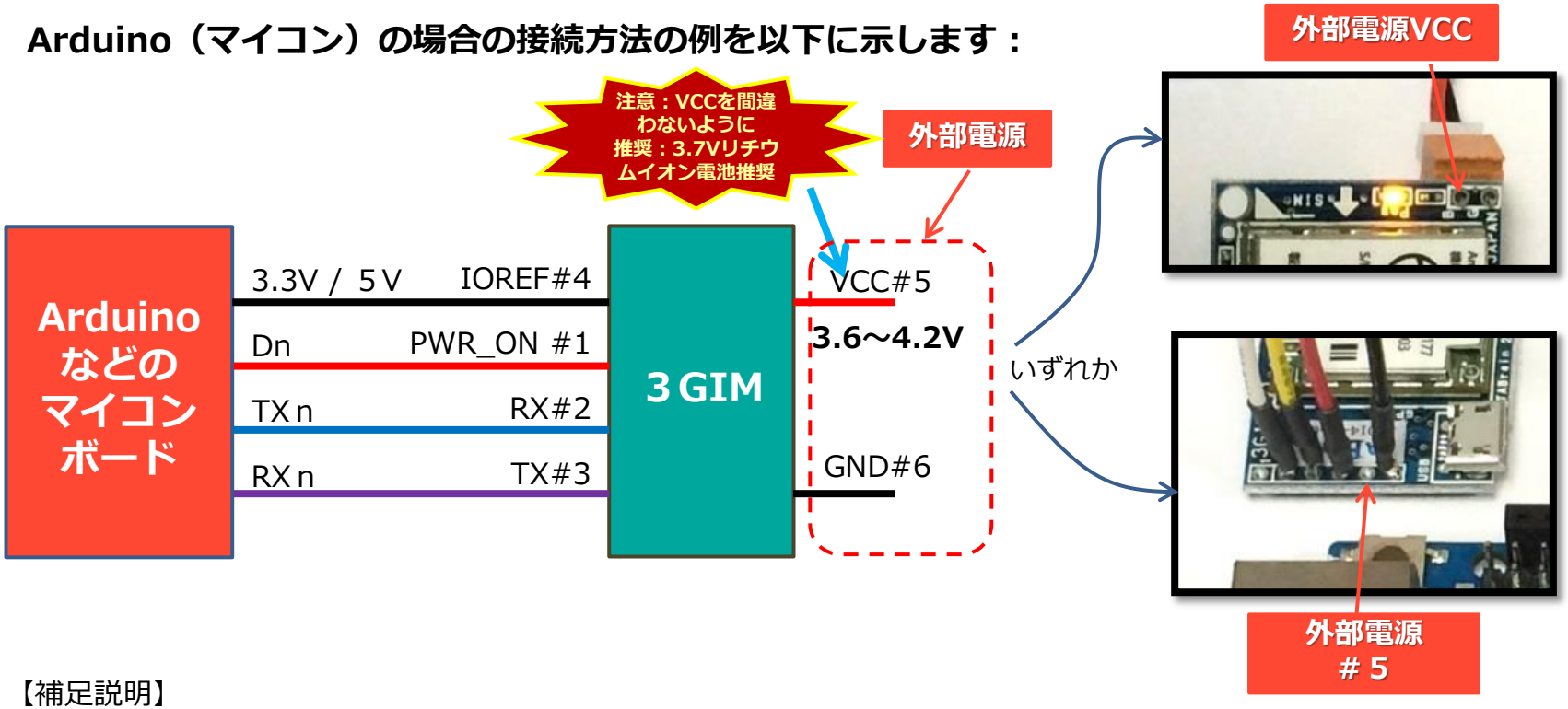
4. 回路図は、オープンソースとして公開します。

5. GPS取得は、電波障害が少ない野外などで行ってください。また初回GPS取得時では、特にPCなどの電波障害を避けて、ご利用ください。（USBケーブルを長いものを使ってPC本体から離してご利用頂くなど）

   （初回のGPS取得は、数分ほど時間が掛かります）

# 【参考】Arduinoで動かす配線・接続

- **Arduino（マイコン）の場合の接続方法の例を以下に示します：**




【補足説明】

- デジタル出力Dn（#1接続）をLOWにすることで、3GIMの電源をHIGHにします。なお、3GIMの初回立ち上げには約30秒以上かかります。Dn（#1接続）を解放していると、常に通信モジュールに電源が供給された状態となります。
  ※一度電源を入れると、初期立上げ以降、コマンド操作での待機時間は不要となります。
- IOREFピンには、使用するArduinoのロジック電圧(3.3V or 5V)を接続します。
  （Arduino から直接 電源を取る事ができます）
- UARTはクロスで接続します（TX/RXを交差させて接続します）

もくじ



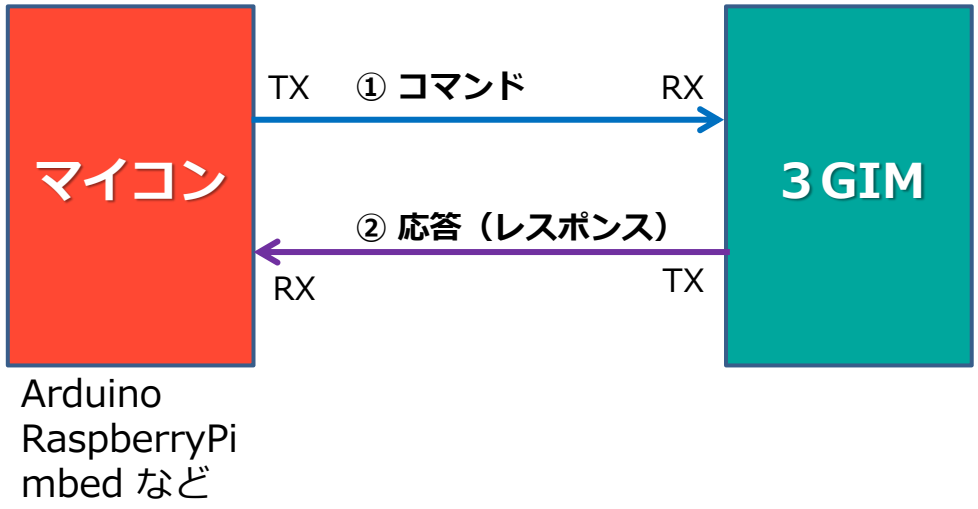
# 第2章
# UARTコマンドインタフェース

# 1．UART送受信インタフェースの概要

## ■UARTによる送信（コマンド）と受信（応答：レスポンス）との関係

- 外部(マイコン側)と３GIMとの通信は、UARTを通じて行います。
- マイコン側から**コマンド**を送信し、３GIM側で受信します。
- つぎに３GIM側から**応答（レスポンス）**を送信し、マイコン側で受信して、コマンド制御を終了します。
- つまりUART送受信の一連の処理は、マイコン側から３GIM側へのコマンド送信と、３GIM側からマイコン側への応答（レスポンス）送信で、１つの**シーケンス**として完結します。
  - コマンドおよび応答（レスポンス）は、改行コード('¥n')で終端します。

（Arduino IDEのシリアルモニタ画面で直接やりとりされる場合には、改行選択メニューで「**CRおよびLF**」を選択してください）



# 2．3GIMコマンド一覧

3GIMのUART経由で利用できるコマンド（3GシールドV2と同一インタフェース）を下表に示します

| No | 分類 | 機能 | コマンド | 頁 | 機能概要 | 補足 |
|---|---|---|---|---|---|---|
| 1 | System | Version | $YV | P.18 | gw3gアプリのバージョン情報の取得 | |
| 2 | | RSSI | $YR | P.20 | 電波受信強度(RSSI)の取得 | |
| 3 | | Serice | $YS | P.22 | 利用可能サービスの取得 | |
| 4 | | IMEI | $YI | P.24 | IMEIの取得 | |
| 5 | | LED | $YL | P.26 | LED（RUN）の状態の取得、設定 | |
| 6 | | Baudrate | $YB | P.29 | UARTの通信速度の変更 | gw3gアプリのリセット後に有効 |
| 7 | | Reset | $YE | P.30 | IEMのリセット | |
| 8 | | Time | $YT | P.32 | 時間の取得 | |
| 9 | | Airplane mode | $YP | P.34 | IEMのエアプレーン（機内）モードの切り替え | R1.3で追加 |
| 10 | | Do Command | $YD | P.36 | 暗号化されたコマンドを実行する | R2.0で追加、プロファイル設定のみ |
| 11 | SMS | Send | $SS | P.39 | SMSの送信 | |
| 12 | | Receive | $SR | P.41 | SMSの受信 | |
| 13 | | Check | $SC | P.43 | SMS着信の有無チェック | |
| 14 | GPS | GPS | $LG | P.46 | 位置情報の取得 | GPSを利用 |
| 15 | Web | Get | $WG | P.49 | GETリクエストの送出、レスポンスの取得 | ヘッダ指定可(R2.0から) |
| 16 | | Post | $WP | P.51 | POSTリクエストの送出、レスポンスの取得 | |
| 17 | TCP/IP | Read | $TR | P.54 | TCP/IPコネクションからのデータからの読み出し | R2.0からバイナリデータも取扱可 |
| 18 | | Write | $TW | P.55 | TCP/IPコネクションへのデータの書き込み | 同上 |
| 19 | | Connect | $TC | P.56 | TCP/IPコネクションの接続 | |
| 20 | | Disconnect | $TD | P.57 | TCP/IPコネクションの切断 | |
| 21 | | Status | $TS | P.58 | TCP/IPコネクションの状態の取得、設定 | |
| 22 | | Get sockname | $TN | P.59 | ソケットのIPアドレスとポート番号を取得 | 接続時のみ有効 |
| 23 | Profile | Set | $PS | P.63 | デフォルトプロファイル番号の設定 | |
| 24 | | Read | $PR | P.65 | デフォルトプロファイル番号の取得、指定プロファイルの取得 | |
| 25 | | Reset | $PE | P.67 | 指定プロファイルのリセット（クリア） | |
| 26 | Storage | Write | $RW | P.70 | IEM内のストレージへデータを書き込む | 同上 |
| 27 | | Read | $RR | P.72 | IEM内のストレージからデータを読み出す | R2.0で追加 |

※３GIMからの応答（レスポンス）は、各コマンドの機能紹介にて説明していきます。

# 3．3GIMのインタフェース形式（共通事項）

## ■コマンドの指定表示形式

$XX 引数1 引数2 …¥n　　※ここでの「XX」はコマンド名です。

引数は1つ以上の半角スペースで区切る。引数には制御コードは含まないでください。

（制御文字を含む引数の指定では、$文字エスケープシーケンスを使用してダブルクォートで囲む）

注意：[　] （カギ括弧）表記は、オプション（省略可）のもので、実際には記述は不要です。

## ■応答（レスポンス）結果表示形式

$XX=OK 【結果】¥n　　※ここでの「XX」はコマンド名です。

【結果】が複数行になる場合は結果部分全体を"で囲みます。

$XX=NG エラーコード 【付加情報】¥n

エラーコードは別途定義する1～3桁の数字となります。

※ 【結果】と【付加情報】はオプションです。

**【補足1】** 3GIMに電源供給して約30秒ほど経たないとファームウェアが立ち上がりません。立ち上げ時の最初には、特殊文字および以下の応答（レスポンス）が返信されます。これらは読み飛ば（無視）して処理してください。

Hello, I'm gw3g(Ver 2.0)

**【補足2】** コマンドが間違った場合の応答（レスポンス）は、以下のようになります。

$=NG 10

## ■コマンドパラメータの特殊文字の表現形式

‘$’ 文字に

| |
|---|
| $t : TAB(0x09) |
| $r : CR(0x0d) |
| $n : NL(0x0a) |
| $" : "そのもの |
| $$ : $そのもの |
| $xhh または $Xhh : 16進数hh |

な文字（コード）を表現します。具体的には下記の通りとなります。

例えば、下記のように使用する：

HTTPヘッダの例　　"Content-Type: text/csv**$r$n**"

# 4．3GIMのインタフェース形式（補足事項①）

## ■処理に時間が掛かるコマンドの途中応答（レスポン）の出力（途中ステータス）について

処理に時間が掛かる以下のコマンドでは、「>」文字を使った文字列が、途中経過の表示（**途中ステータス**）として応答（レスポンス）として出力されます。

LG, WG, WP, TC, TR

例えば、以下の応答が返ってきます。（$WGの場合の**途中ステータス**）

```
$WG http://3gsa.org/
>W*=STARTING
>W*=GETHOSTBYNAME
>W*=CONNECT
>W*=SENDREQUEST
>W*=READRESPONSE
>WG=CONTENT_TYPE text/html
>WG=READ(578Byte)
>WG=READ(445Byte)
```

※応答（レスポンス）結果だけを知りたい場合は、途中ステータスの「>」で始める行を無視するプログラムが必要です。

つまり、結果表示「$」で始まる行以外は読み捨てるなどが必要です。

※特に応答までの時間が掛る「**$LG**」による初回のGPS取得の場合には、何度かコマンドを実行してみる必要もあります。

# 4．３GIMのインタフェース形式（補足事項②）

## ■電源供給後に3GIMが利用可能となった際に出力される内容

3GIMに電源を供給し、コマンド送信した後、下記の応答（レスポンス）メッセージが返ってくる場合があります。

```
Hello, I'm gw3g(Ver 2.0)
```

※下線部は、3GIMのファームウェアのバージョンを示す

これは、３GIMのファームウェアの立ち上げ時に出てくるメッセージとなります。つまり電源投入後、待機時間（推奨35秒）よりも短い時間でコマンド送信したときなどに応答する文字列となります。

このメッセージは、特に問題が出たことではなく、無視できる応答（レスポンス）です。

## ■応答（レスポンス）がない場合についての処理

正常な実行中に、応答（レスポンス）がない場合は、異常時（例えば、IEMがハングアップした等）となります。通常は起こりえませんが、IEMがハングアップした時などの異常時への対応としてタイムアウト処理などが必要となります。

もくじ



# 第３章
# ３GIMコマンド・応答（レスポン）


# 1．System関連

# 1．SYSTEM VERSION ①

## ■ IEM（通信）モジュールに設定されたファームウェアgw3gのバージョン取得

通信モジュール（IEMモジュール）に設定されているファームウェアのバージョンを取得する

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | System | | |
| 機能名 | VERSION | gw3gアプリケーションのバージョンを取得する | |
| コマンド形式 | | $YV¥n | |
| 引数 | - | | |
| 応答値 | 【正常時】 | $YV=OK version¥n | |
| | version | "9.9"形式のバージョン（整数桁がメジャ番号、小数以下がマイナ番号）※ | 必ず正常終了する |
| | 【エラー時】 | | |
| 前提条件 | | | |
| 補足事項 | | | |

※最新バージョンは、「2.0」となる。(2015.04現在)

# 1．SYSTEM VERSION ②

## ■事例：バージョン取得サンプルプログラム

■Arduinoでのサンプルスケッチ

```
#include <SoftwareSerial.h>
SoftwareSerial iemSerial(4,5);
void setup(){
  Serial.begin(9600);
  iemSerial.begin(9600);
  delay(35000);
  Serial.println("begin System Version");
  iemSerial.print("$YV¥n");
  iemSerial.listen();
  while(!iemSerial.available());
  String rt = iemSerial.readStringUntil('¥n');
 Serial.println(rt);
 Serial.println("end");
}
void loop() {}
```

通信速度設定

コマンド送信

応答受信

【補足】ここでの呼出し関数群について
・SoftwareSerial は、ソフトシリアル通信利用宣言
・iemSerial.begin(通信速度)は、 ボーレート宣言
・delay(num)は、 numミリ秒の待機時間
・iemSerial.listen()は、受信状態占有関数
・iemSerial.available()は、受信バイト数を返す
・iemSerial.readStringUntil(‘¥n’)は、
　　リターン値までの文字列読込み関数

■実行モニタ画面（サンプル）

```
begin System Version
$YV=OK 2.0
end
```

応答受信

返信が無い場合は、以下のようなことが考えられる
1）通信モジュールに電源が入ってない
2）通信ボーレートが間違っている
3）配線（特にUARTのTxとRxの接続）が間違っている

# 2．SYSTEM RSSI ①

## ■電波受信強度（RSSI）の取得

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | System | | |
| 機能名 | RSSI | 現在のRSSI値を取得する | |
| コマンド形式 | | $YR¥n | |
| 引数 | - | | |
| 応答値 | 【正常時】 | $YR=OK rssi¥n | |
| | rssi | 電波強度[dBm] | rssiは0未満のマイナス値 |
| | 【エラー時】 | $YR=NG errno¥n | |
| | errno | 101：電波強度が取得できない | |
| 前提条件 | | | |
| 補足事項 | | | |

**RSSI**とは、無線通信機器が受信する信号の強度を測定するための回路または信号のこと。Received Signal Strength Indication, Received Signal Strength Indicator別名：受信信号強度のこと。

RSSI値は、マイナス値で、以下の状況下となる
-125の場合には、アンテナが接続されていないとき
-124～-100 の場合は、電波受信状態が悪い状況
-99～　の場合は、電波受信状態が良い状況

# 2．SYSTEM RSSI ②

## ■事例：電波受信強度（RSSI）を取得サンプルプログラム

■Arduinoでのサンプルスケッチ

```
#include <SoftwareSerial.h>
SoftwareSerial iemSerial(4,5);
void setup(){
  Serial.begin(9600);
  iemSerial.begin(9600);
  delay(35000);
  Serial.println("begin SYSTEM RSSI");
  iemSerial.print("$YR¥n");
  iemSerial.listen();
  while(!iemSerial.available());
  String rt = iemSerial.readStringUntil('¥n');
  Serial.println(rt);
  Serial.println("end");
}
void loop() {}
```

通信速度設定

コマンド送信

■実行モニタ画面（アンテナ正常接続）

```
begin SYSTEM RSSI
$YR=OK -86
end
```

応答受信

■実行モニタ画面（アンテナ不接続）

```
begin SYSTEM RSSI
$YR=OK -125
end
```

応答受信

RSSIの感度が悪い場合
1）通信モジュールとアンテナのコネクタが正しく接続されていない
2）アンテナとケーブル・コネクタのネジ部が緩んでいる
3）電波状態が悪い屋内の壁・天井・床などで閉ざされたところにある

# 3．SYSTEM SERVICE ①

## ■SIMカードによる通信サービス状況を取得

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | System | | |
| 機能名 | SERVICE | 現在利用できる通信サービスを取得する | |
| コマンド形式 | | $YS¥n | |
| 引数 | - | | |
| 応答値 | 【正常時】 | $YS=OK serice¥n | 必ず成功する |
| | service | 0：サービス利用不可 | |
| | | 1：パケット通信(PS)のみ利用可 | |
| | | 2：音声通信(CS)のみ利用可 | |
| | | 3：パケット通信(PS)および音声通信(CS)の両方が利用可 | |
| | 【エラー時】 | – | |
| 前提条件 | ① | あらかじめSIMカードが装着されていること | SIMカードがないと常に結果として0が返る |
| 補足事項 | | | |

**3GIM**で利用できる**SIMカード**は、以下のインターネット上の技術サポート（Wikiページ）サイトに掲載しています。

http://a3gs.wiki.fc2.com/wiki/SIMカード情報

# 3．SYSTEM SERVICE ②

## ■事例：SIMカードによる通信サービス状況を取得サンプルプログラム

■Arduinoでのサンプルスケッチ

```
#include <SoftwareSerial.h>
SoftwareSerial iemSerial(4,5);
void setup(){
  Serial.begin(9600);
  iemSerial.begin(9600);
  delay(35000);
  Serial.println("begin SYSTEM Service");
  iemSerial.print("$YS¥n");
  iemSerial.listen();
  while(!iemSerial.available());
  String rt = iemSerial.readStringUntil('¥n');
 Serial.println(rt);
 Serial.println("end");
}
void loop() {}
```

通信速度設定

コマンド送信

■実行モニタ画面（パケット通信のみの利用の場合）

```
begin SYSTEM Service
$YS=OK 1
end
```

応答受信

# 4．SYSTEM IMEI ①

■通信モジュールのID（固有）番号（IEM）を取得

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | System | | |
| 機能名 | IMEI | IEMのIMEIを取得する | |
| コマンド形式 | | $YI¥n | |
| 引数 | - | | |
| 応答値 | 【正常時】 | $YI=OK imei¥n | |
| | imei | 15桁の数字 | |
| | 【エラー時】 | $YI=NG errno¥n | |
| | errno | ISHELL_GetDeviceInfoEx()の戻り値 | |
| 前提条件 | | | |
| 補足事項 | | | |

**IMEI**は「国際移動体装置識別番号（端末識別番号）」を意味する英語"International Mobile Equipment Identifier"の略

# 4．SYSTEM IMEI ②

## ■事例：通信モジュールのID（固有）番号（IEM）を取得サンプルプログラム

■Arduinoでのサンプルスケッチ

```
#include <SoftwareSerial.h>
SoftwareSerial iemSerial(4,5);
void setup(){
  Serial.begin(9600);
  iemSerial.begin(9600);
  delay(30000);
  Serial.println("begin SYSTEM IMEI");
  iemSerial.print("$YI¥n");
  iemSerial.listen();
  while(!iemSerial.available());
  String rt = iemSerial.readStringUntil('¥n');
 Serial.println(rt);
 Serial.println("end");
}
void loop() {}
```

通信速度設定

コマンド送信

■実行モニタ画面（正常時）

```
begin SYSTEM IMEI
$YI=OK 356423042101234
end
```

応答受信

※このID番号の数値はサンプルです

■実行モニタ画面（エラー時）

```
begin SYSTEM IMEI
$=NG 10
end
```

※コマンドが正しく読み込めなかった場合

IMEIの一部番号

# 5．SYSTEM LED ①

## ■ 3GIM上のLEDの点滅設定および状態取得

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | System | | |
| 機能名 | LED | IEMのLED（RUN）ピンの状態取得・設定を行う | |
| コマンド形式 | 状態取得 | $YL¥n | |
| | 設定 | $YL status¥n | |
| 引数 | status | ONにするか(1の時)、OFFにするか(0の時) | |
| 応答値 | 【正常時】 | $YL=OK status¥n | |
| | status | 本コマンド実行後のLED状態（0:OFF／1:ON） | |
| | 【エラー時】 | $YL=NG errno¥n | |
| | errno | 191：status引数の値がおかしい | |
| 前提条件 | | | |
| 補足事項 | ① | 本関数で扱うLEDは以下の位置となる。 | |
| | ② | 3GIMでは、LEDはIEM脇に配置されている<br>（V1.0では点灯時でも暗いので注意） | |

# 5．SYSTEM LED ②

## ■事例：LED状態取得とLEDの点滅１０回繰り返しのサンプルプログラム

■Arduinoでのサンプルスケッチ

```
#include <SoftwareSerial.h>
SoftwareSerial iemSerial(4,5);
void setup(){
  Serial.begin(9600);
  iemSerial.begin(9600);
  delay(35000);
  Serial.println("begin SYSTEM LED");
  iemSerial.print("$YL¥n");
  while(!iemSerial.available());
  String rt = iemSerial.readStringUntil('¥n');
  Serial.print(rt);
  for(int i=0; i<10; i++) {
    iemSerial.print("$YL 1¥n");
    delay(500);
    iemSerial.print("$YL 0¥n");
    delay(500);
  }
  Serial.print("¥r¥nend");
}
void loop() {}
```

点滅を10回繰り返す

■実行モニタ画面（正常時）

```
begin SYSTEM LED
$YL=OK 0
end
```

# 6．SYSTEM BAUDRATE ①

## ■３GIMのUART（通信ポート）の通信速度（ボーレート）取得確認と設定

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | System | | |
| 機能名 | BAUDRATE | UARTの通信速度(ボーレート)の取得・設定を行う | |
| コマンド形式 | 取得 | $YB¥n | |
| | 設定 | $YB baudrate¥n | |
| 引数 | baudrate | 設定するボーレート<br>(1200/2400/4800/9600/19200/38400/57600/115200) | |
| 応答値 | 【正常時】 | $YB=OK baudrate¥n | |
| | baudrate | 本コマンド実行後のボーレート | 取得時のみbaudrateが出力される |
| | 【エラー時】 | $YB=NG errno¥n | |
| | errno | 111：引数baudrateがおかしい | |
| | | 112：内部エラー | |
| 前提条件 | ① | 指定するボーレートで正しく動作することを確認しておくこと | ボーレートの目安を参照のこと |
| 補足事項 | ① | 本コマンドで設定したボーレートは、IEMをリセットした後に有効となる。 | RESETコマンド($YE)を実行する |
| | | （リセットするまでは、現在のボーレートは変更されない） | |
| | ② | 本コマンドの実行には十分留意すること。設定したボーレートが不適切な場合は、gw3gアプリがUART経由で利用できなくなる。 | |
| | ③ | 内部処理として、gw3g設定ファイル(gw3g.dat)を書き換える | |

【注意事項】
① 現状のボーレートと、変更後のボーレートは、常に把握した上でこのコマンドを使うこと
（分からくなる/通信できなくなると一つ一つボーレートを試して探り当てる必要がある）
② ソフトウェアシリアル通信状態では、マイコンボードの不安定さもあり、ハードウェアシリアル通信より低速度になる
（ハードウェアシリアル通信だと５７６００bpsでも大丈夫だが、ソフトウェアシリアル通信だと9600bps以下が推奨）

# 6．SYSTEM BAUDRATE ②

## ■事例：３GIMのUARTのボーレート取得確認サンプルプログラム

■Arduinoでのサンプルスケッチ

```
#include <SoftwareSerial.h>
SoftwareSerial iemSerial(4,5);
void setup(){
  Serial.begin(9600);
  iemSerial.begin(9600);
  delay(30000);
  Serial.println("begin SYSTEM Baudrate");
  iemSerial.print("$YB¥n");
  while(!iemSerial.available());
  String rt = iemSerial.readStringUntil('¥n');
  Serial.print(rt);
  Serial.print("¥r¥nend");
}
void loop() {}
```

この通信速度と一致する

■実行モニタ画面（正常時）

```
begin SYSTEM Baudrate
$YB=OK 9600
end
```

ボーレート設定での注意点
１）ボーレート設定を変更すると、改めてシリアル通信速度も変更が必要となります。
２）ボーレート変更設定した場合には、その情報は控えておいてください。

# 7．SYSTEM RESET ①

■ **IEMをリセット（Shutdown）するコマンド**

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | System | | |
| 機能名 | RESET | IEMをリセットする | |
| コマンド形式 | ソフトリセット | $YE¥n | |
| | 指定リセット | $YE level¥n | |
| 引数 | level | リセットのレベル（0: ソフトリセット、1:ハードリセット） | |
| 応答値 | 【正常時】 | $YE=OK level¥n | |
| | level | 引数と同じ | |
| | 【エラー時】 | $YE=NG errno¥n | |
| | errno | ISHELL_Reset()の戻り値 | |
| 前提条件 | | | |
| 補足事項 | ① | リセットには40秒程度の時間が掛かる。再起動するまでIEMは利用できない。 | |
| | ② | IEMがハングアップした時は、3GIM基板#1による電源OFF/ONを使い、強制的にリセットを行うこと | |
| | ③ | 実装上の理由で、ハードリセットはソフトリセットと同じ動作となっている。 | |

# 7．SYSTEM RESET ②

## ■事例：リセットのサンプルプログラム

■Arduinoでのサンプルスケッチ

```
#include <SoftwareSerial.h>
SoftwareSerial iemSerial(4,5);
void setup(){
  Serial.begin(9600);
  iemSerial.begin(9600);
  delay(30000);
  Serial.println("begin SYSTEM Baudrate");
  iemSerial.print("$YE¥n");
  while(!iemSerial.available());
  String rt = iemSerial.readStringUntil('¥n');
  Serial.print(rt);
  Serial.print("¥r¥nend");
}
void loop() {}
```

■実行モニタ画面（正常時）

```
begin SYSTEM Baudrate
$YE=OK 1
end
```

# 8．SYSTEM TIME ①

## ■通信モジュールの取得した時間取得

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | System | | |
| 機能名 | System Time | 日時時間の取得 | |
| コマンド形式 | モード取得 | $YT¥n | |
| 引数 | | | |
| 応答値 | 【正常時】 | $YT=OK datetime¥n | |
| | datetime | 出力例として<br>$YT=OK 2015/03/26 15:52:23<br>などのように「年（4バイト）’/’月（2バイト）’/’日（2バイト）’ ’（スペース1バイト）時（2バイト）’:’分（2バイト）’:’秒（2バイト）」でリターン値が返ってくる。 | |
| | 【エラー時】 | $YP=NG¥n | |
| 前提条件 | | アンテナ接続と正しいSIMカードの設定で正確な時刻が取得できる | |
| 補足事項 | | 通信モジュールの個体差があり、数分ほど電源供給していないと正しく日時が取得できない。 | |

# 8．SYSTEM TIME ②

## ■ 事例：日時取得表示のサンプルプログラム

■Arduinoでのサンプルスケッチ

```
#include <SoftwareSerial.h>
SoftwareSerial iemSerial(4,5);
void setup(){
  Serial.begin(9600);
  iemSerial.begin(9600);
  delay(30000);
  Serial.println("begin System Time");
  iemSerial.print("$YT¥n");
  while(!iemSerial.available());
  String rt = iemSerial.readStringUntil('¥n');
  Serial.print(rt);
  Serial.print("¥r¥nend");
}
void loop() {}
```

■実行モニタ画面（正常時）

```
begin AirPlane Mode
$YT=OK 2015/03/26 15:50:53
end
```

【補足】取得した時間が間違っている場合
1）正しいSIMカードが設定されているかを確認
2）アンテナ接続が正しく接続されているかを確認
3）アンテナ感度が良い環境かどうかを確認（参照：＄YR)
4）正しい時間取得までに数分から10.分ほどかかる場合がある

# 9．AIRPLANE MODE ①

■ エアプレーンモード（機内モード）の取得・設定

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | System | | |
| 機能名 | Airplane mode | IEMのエアプレーン（機内）モードを切り替える | |
| コマンド形式 | モード取得 | $YP¥n | |
| | モード切替 | $YP mode¥n | |
| 引数 | mode | 設定するモード（0: 通常モード、1:エアプレーンモード） | |
| 応答値 | 【正常時】 | $YP=OK mode¥n | |
| | mode | 設定後のモードを返す<br>０：通常モード<br>１：エアプレーン（機内）モード | |
| | 【エラー時】 | $YP=NG errno¥n | |
| | errno | 未定 | |
| 前提条件 | ① | gw3gのバージョンがR1.3以降のみで利用できる | |
| 補足事項 | ① | エアプレーンモードの時は、SMSを含めて通信操作は一切行えないが、消費電力は少なくなる。 | |

# 9．AIRPLANE MODE ②

## ■ 事例：エアプレインモード値の取得サンプルプログラム

■Arduinoでのサンプルスケッチ

```
#include <SoftwareSerial.h>
SoftwareSerial iemSerial(4,5);
void setup(){
  Serial.begin(9600);
  iemSerial.begin(9600);
  delay(30000);
  Serial.println("begin AirPlane Mode");
  iemSerial.print("$YP¥n");
  while(!iemSerial.available());
  String rt = iemSerial.readStringUntil('¥n');
  Serial.print(rt);
  Serial.print("¥r¥nend");
}
void loop() {}
```

■実行モニタ画面（正常時）

```
begin AirPlane Mode
$YP=OK 0
end
```

# 10. DO COMMAND ①

本コマンドはユーザ独自では利用不可

■ 事例：新しいSIMプロファイルの設定

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | SYSTEM | | |
| 機能名 | Do command | 暗号化された引数（APN情報）で$PWコマンドを実行する | |
| コマンド形式 | 実行 | $YD password "encrypted-data"¥n | |
| 引数 | password | パスワード（英数字または空白・ダブルクォートを除く記号からなる文字列） | 最大16バイト、ASCII文字のみ |
| | encrypted-data | 暗号化されたコマンド文字列 | 最大256バイト、ASCII文字のみ |
| 応答値 | 【正常時】 | $YD=OK¥n | 正常にコマンドを復号し、実行できた時 |
| | 【エラー時】 | $YD=NG errno¥n | コマンドを実行できなかった時 |
| | errno | 151：パスワードエラー | |
| | | 152：コマンド文字列エラー | |
| | | 153：内部エラー | |
| 前提条件 | ① | gw3g R2.0以降のみで利用できる | |
| 補足事項 | ① | 暗号・復号方式は、「MD5+ARC4」を採用する。 | |

※ 本コマンドは、新たなSIMプロファイルを設定するためのコマンドで、予めSIMカードのプロファイル情報（APN情報、ID、PW）が必要 です。
<本件は、有償にてメール申込み対応： info@tabrain.jp まで>

# 10. DO COMMAND ②

本コマンドはユーザ独自では利用不可

## ■事例：新たなSIMプロファイルの設定サンプルプログラム

＜本コマンドは、予めメール：info@tabrain.jp にてSIMプロファイルを申請し、パスワードと暗号化された設定情報をもって実行（有償サービス）＞

■Arduinoでのサンプルスケッチ

```
#include <SoftwareSerial.h>
SoftwareSerial iemSerial(4,5);
void setup(){
  Serial.begin(9600);
  iemSerial.begin(9600);
  delay(30000);
  Serial.println("begin Do Command");
  iemSerial.print("$YD PW "*****"¥n");
  while(!iemSerial.available());
  String rt = iemSerial.readStringUntil('¥n');
  Serial.print(rt);
  Serial.print("¥r¥nend");
}
void loop() {}
```

■実行モニタ画面（正常時）

```
begin Do Command
$YD=OK
end
```

■返却（パスワード抜け）

```
begin Do Command
$YD=NG  151 password
end
```

■実行モニタ画面（パスワード間違い）

```
begin Do Command
$YD=NG 221 AUTH
end
```

■実行モニタ画面（暗号データ間違い）

```
begin Do Command
$YD=NG 221 APN
end
```

■実行モニタ画面（暗号データ抜け）

```
begin Air Do Command
$YD=NG 152 encrypted-command
end
```


# 2．SMS関連

# 1．SMS SEND ①

## ■SMS(ショートメッセージ）の送信

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | SMS | | |
| 機能名 | SEND | SMSを送信する | |
| コマンド形式 | | $SS msn "message" [encode]¥n | encode=ASCII と同じ |
| 引数 | msn | 送信先の電話番号（ハイフォン無しの数字のみで指定） | |
| | message | 送信するメッセージ（制御文字は使用不可、日本語はUNICDEで記述） | 「"」は「¥"」として記述、最大100バイトまで |
| | encode | "ASCII" または "UNICODE"のいずれか | |
| 応答値 | 【正常時】 | $SS=OK¥n | |
| | 【エラー時】 | $SS=NG errno ..¥n　または　$SS=NG errtype errcode¥n | |
| | errno | 401：引数指定エラー | |
| | | 402：BUSYエラー（すでにSMSを送信中） | |
| | errtype | AEESMS_GETERRORTYPE()の応答値 | |
| | errcode | AEESMS_GETERROR()の応答値 | |
| 前提条件 | ① | 音声サービス(SMS含む)が利用できる状態であること。 | |
| 補足事項 | ① | 文字コードがASCIIの場合でも、SMSとして利用できない文字が存在する。 | |

※ SMS送信の場合、確認のための**送信ステータス**を表示します。（参照：P14）
※ SMS（ショートメッセージ）の送信の応答性は、必ずしも即時性があるとは限りません。（タブレイン調査）
※ SMS （ショートメッセージ）は、NTTドコモでは1件あたり3円（税別）となっていますので、大量送信では気を付けるようにしてください。

# 1．SMS SEND ②

## ■事例：SMS(ショートメッセージ）送信のプログラム

■Arduinoでのサンプルスケッチ

```
#include <SoftwareSerial.h>
SoftwareSerial iemSerial(4,5);
void setup(){
  Serial.begin(9600);
  iemSerial.begin(9600);
  delay(30000);
  Serial.println("begin SMS Send ");
  iemSerial.print("$SS 08012345678 ¥"Hello! SMS Send¥" ASCII¥n");
  String rt="";
  do{
    while(!iemSerial.available());
    rt = iemSerial.readStringUntil('¥n');
    Serial.print(rt);
  } while(!rt.startsWith("$SS="));
  Serial.print("¥r¥nend");
}
void loop() {}
```

＄SS=OKまたは＄SS=NGでない場合繰り返し

■実行モニタ画面（正常時）

```
begin SMS Send
>SS=SENDING AS ASCII
$SS=OK
end
```

途中ステータス

# 2．SMS RECEIVE ①

## ■SMS(ショートメッセージ）の受信

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | SMS | | |
| 機能名 | RECEIVE | 受信したSMSを読み出す | |
| コマンド形式 | | $SR¥n | |
| 引数 | － | | |
| 応答値 | 【正常時】 | $SR=OK msn "message"¥n | 「"」は「¥"」として記述 |
| | msn | 受信したSMSの送信元の電話番号（ハイフォン無し） | 最大11バイト |
| | message | 受信したSMSのメッセージ | ASCIIまたはUNICODE、最大100バイト |
| | 【エラー時】 | $SR=NG errno¥n | |
| | errno | 412：SMSを受信していない | |
| | | | |
| 前提条件 | ① | 音声サービス(SMS含む)が利用できる状態であること。 | |
| 補足事項 | ① | 本関数の実行により、IEMのLEDピンはHIGHに変更される。 | |

# 2．SMS RECEIVE ②

## ■事例：SMS（ショートメッセージ）の受信サンプルプログラム

### ■Arduinoでのサンプルスケッチ

```
#include <SoftwareSerial.h>
SoftwareSerial iemSerial(4,5);
void setup(){
  Serial.begin(9600);
  iemSerial.begin(9600);
  delay(30000);
  Serial.println("begin SMS Receive ");
  iemSerial.print("$SR¥n");
  String rt="";
  while(!iemSerial.available());
  rt = iemSerial.readStringUntil('¥n');
  Serial.print(rt);
  Serial.print("¥r¥nend");
}
void loop() {}
```

### ■実行モニタ画面（正常時）

```
begin SMS Receive
$SR=OK  08012345678 " Hello! SMS Send "
end
```

# 3．SMS CHECK ①

■SMS（ショートメッセージ）の受信確認

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | SMS | | |
| 機能名 | CHECK | SMSを受信しているかどうかをチェックする | |
| コマンド形式 | | $SC¥n | |
| 引数 | － | | |
| 応答値 | 【正常時】 | $SC=OK rtn¥n | |
| | rtn | 0： SMSを受信していない時 | |
| | | 1： SMSを受信している時 | |
| 前提条件 | | | |
| 補足事項 | ① | 本関数の実行により、IEMのLEDピンの状態は維持される。 | |

# 3．SMS CHECK ②

## ■事例：SMS（ショートメッセージ）受信サンプルプログラム

### ■Arduinoでのサンプルスケッチ

```
#include <SoftwareSerial.h>
SoftwareSerial iemSerial(4,5);
void setup(){
  Serial.begin(9600);
  iemSerial.begin(9600);
  delay(30000);
  Serial.println("begin SMS Check ");
  iemSerial.print("$SC¥n");
  while(!iemSerial.available());
  String  rt = iemSerial.readStringUntil('¥n');
  Serial.print(rt);
  Serial.print("¥r¥nend");
}
void loop() {}
```

### ■実行モニタ画面（正常時）

```
begin SMS Check
$SC=OK  0
end
```

未受信

```
begin SMS Check
$SC=OK  1
end
```

受信

※ SMS受信は、即時性がないため、しばらく何回か受信確認が必要となる場合があります。


# 3．GPS関連

# 1．LOCATION GPS ①

■ GPS(全地球測位システム）の取得コマンド

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | GPS | | |
| 機能名 | GPS START | 測位を行う。 | |
| コマンド形式 | | $LG method¥n | |
| 引数 | method | 測位の方法（下記のいずれかを指定） | |
| | | MSBASED: GPSで測位、GPSが利用できない時は3Gネットワークを利用 | |
| | | MSASSISTED: 3Gネットワークを利用して測位 | |
| | | STANDALONE: GPS単体で測位 | |
| 応答値 | 【正常時】 | $LG=OK latitude longitude¥n | |
| | latitude | 緯度（北緯、9.99999形式、ただし桁数は場合により可変） | |
| | longitude | 経度（東経、9.99999形式、ただし桁数は場合により可変） | |
| | 【エラー時】 | $LG=NG errno¥n | |
| | errno | 501：引数指定エラー | |
| | | 508：GPS測位エラー | |
| | | 509：BUSYエラー（すでに測位中） | |
| 前提条件 | ① | GPSアンテナが正しく装着されてること | |
| 補足事項 | ① | 測位には、初回には数分以上の時間がかかる場合がある。 | |
| | ② | AGPSサーバとして、Googleのロケーションサーバを利用する。 | |
| | ③ | 本コマンドの初回GPS取得までには数分から10分ほど時間が掛かる。従って、本コマンドの実行では結果が返ってくるまで、何度か繰り返し実行か、待機する必要がある。 | |

# 1．LOCATION GPS ②

## ■事例：GPS取得プログラム

■Arduinoでのサンプルスケッチ

```
#include <SoftwareSerial.h>
SoftwareSerial  iemSerial(4, 5);
const unsigned long baudrate = 9600;

void setup() {
  Serial.begin(baudrate);
  iemSerial.begin(baudrate);
    pinMode(7,OUTPUT);
  digitalWrite(7,HIGH);
  Serial.println("begin Location GPS");
  delay(35000);
}

void loop() {
  static boolean sw=true;
  if(sw) {
      iemSerial.write("$LG MSBASED¥n");
      Serial.println("$LG MSBASED");
  }
  sw=false;
  while(!iemSerial.available());
  String rt = iemSerial.readStringUntil('¥n');
  Serial.println(rt);
  if(rt.startsWith("$LG=NG 508")){  sw=true;
  }
  else if ( rt.startsWith("$LG=OK")) {
    Serial.println("end "); while(1);
  }
  if(rt.startsWith("$LG=NG 509")) sw=false;
}
```

■シリアルモニタ画面（正常時応答）

```
begin Location GPS
$LG MSBASED
>LG=START 4
$LG=OK 35.62212345 139.5912345
 end
```

【補足説明：初回のGPS取得できない理由】
※ $LG=NG 508 が時間オーバで返ってくるとき
1）屋内などのGPS衛星がとらえられない
→ なるべく屋外などで電波状態が良い環境で実行
2）数分でGPS取得できない
→ 初回は10.分以上ほどかかる場合もあるので何度か実行
3）アンテナケーブルや電源の不備
・GPSアンテナが正しく接続されていない
・外部電源供給が不足している
4） 3Gと併用してGPS取得できない
→ 正しく稼働するSIMカードが挿入されていること
5）パソコンなどの近くでは電波障害でGPS取得しにくい
→ パソコン本体などから3GIMを離して実行する
6）Baudrate が早くて通信不具合となる
→ 一度4800bpsなどの低速環境で実施してみる


# 4．Web関連

# 1．HTTP GET ①

## ■HTTP GETによるネット接続

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | Web | | |
| 機能名 | GET | HTTP/GETを指定されたURLへ送信して、レスポンスを取得する | |
| コマンド形式 | | $WG url ["header"]¥n | カギ括弧［］は、実際は不要 |
| 引数 | url | GETリクエストを送信するURL（例えば、"http://www.google.co.jp/"等） | URLエンコードされていること |
| | | | 先頭に"http://"または"https://"を含むこと |
| | header | ヘッダ情報（例えば、"Authorization: Basic QWxhZGRpblc2FtZQ==）"等） | ヘッダ部の末尾に改行付きで付与される。$エンコードされていること |
| 応答値 | 【正常時】 | $WG=OK nbytes¥nresponse¥n | |
| | nbytes | レスポンス文字列のバイト数（デコード前のサイズ） | 最大1024 |
| | response | レスポンスの文字列（エンコードされた文字列） | |
| | 【エラー時】 | $WG=NG errno ..¥n | |
| | errno | 301：引数指定エラー | |
| | | 309：BUSYエラー（Web機能を実行中） | |
| | | -534：SIMカードのプロファイル設定エラー | $PS コマンド利用 |
| 前提条件 | ① | パケット通信サービスが利用できる状態であること。 | |
| | ② | ヘッダ情報の指定は、gw3g R2.0以降のみで利用できる | |
| 補足事項 | ① | レスポンスにはヘッダ情報は含まれず、ボディ情報のみが含まれる。 | |
| | ② | レスポンスの文字コードは、urlで指定されたサーバに依存する。 | |

# 1．HTTP GET ②

## ■事例：ネット接続サンプルプログラム

■Arduinoサンプルプログラム

```
#include <SoftwareSerial.h>
SoftwareSerial iemSerial(4,5);
void setup(){
  Serial.begin(9600);
  iemSerial.begin(9600);
  delay(35000);
  Serial.println("begin HTTP GET");
  delay(1000);
 iemSerial.println(" ("$WG http://tabrain.jp/demo/httpGET_test.txt");
 delay(1000);
 unsigned long tm = millis();
 while((millis()-tm<15000)) {
   while(iemSerial.available()) {
      char c= iemSerial.read();
      Serial.print(c);
     tm=millis();
   }
 }
 Serial.print("¥r¥nend");
}
void loop() {}
```

■シリアルモニタ画面（正常時応答）

```
begin HTTP GET
>W*=STARTING
>W*=GETHOSTBYNAME
>W*=CONNECT
>W*=SENDREQUEST
>W*=READRESPONSE
>WG=CONTENT_TYPE text/plain
>WG=READ(44Byte)
$WG=OK 44
Tabrain Web site
Complete access from 3GIM

end
```

■www.tabrain.jp/demo/httpGET_test.txtのファイル内容

サンプルデータ

```
Tabrain Web site
Complete access from 3GIM
```

# 2．HTTP POST ①

## ■HTTP POSTによるネット接続

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | Web | | |
| 機能名 | POST | HTTP/POSTを指定されたURLへ送信して、レスポンスを取得する | |
| コマンド形式 | | $WP url "body" ["header"]¥n | カギ括弧［］は、実際は不要 |
| 引数 | url | POSTリクエストを送信するURL | 最大256バイト（$エンコードされていること） |
| | body | POSTするボディ | 最大1024バイト（〃） |
| | header | ヘッダ情報 | 最大256バイト（〃）、省略可 |
| 応答値 | 【正常時】 | $WP=OK nbytes¥nresponse¥n | |
| | nbytes | レスポンス文字列のバイト数（デコード前のサイズ） | 最大1024バイト |
| | response | レスポンスの文字列（エンコードされた文字列） | |
| | 【エラー時】 | $WP=NG errno ..¥n | |
| | errno※ | 301：引数指定エラー | |
| | | 309：BUSYエラー（Web機能を実行中） | |
| 前提条件 | ① | パケット通信サービスが利用できる状態であること。 | |
| 補足事項 | ① | レスポンスにはヘッダ情報は含まれず、ボディ情報のみが含まれる | |
| | ② | レスポンスの文字コードは、urlで指定されたサーバに依存する。 | |

※「errono」は、WiKiページなどで補足説明していきます。

# 2．HTTP POST ②

■事例：HTTP POSTによるツイート参照

本事例は、後述しています
「3GIMでのツイッター連携使用例」
を参考にしてください。


# 5．TCP/IP関連

# 1．TCP/IP READ

## ■コネクションからのデータ読み込み

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | TCP/IP | | |
| 機能名 | READ | 現在のコネクションからデータを読み出す | ノンブロッキングで動作する |
| コマンド形式 | | $TR maxbytes¥n | |
| 引数 | maxbytes | 読み出すデータの最大長（バイト） | 最大1024 |
| 応答値 | 【正常時】 | $TR=OK nbytes¥ndata¥n | |
| | nbytes | 読み出したデータのバイト数 | |
| | data | 読み出したデータ | gw3g R2.0からバイナリデータも取扱可 |
| | 【エラー時】 | $TR=NG errno ..¥n | |
| | errno | 401：引数指定エラー | |
| 前提条件 | ① | TCP/IPコネクションが確立されていること | |
| 補足事項 | ① | 相手から受信した生のデータをそのまま取得する | |
| | ② | 呼び出された時にIEMに届いているデータを、最大msxbytes分まで読み出す。<br>常にブロッキングせず、データがない時は nbytes=0で直ちに戻る。 | R2.0から常にノンブロッキングで動作 |

# 2．TCP/IP WRITE

■コネクションへのデータ書き出し

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | TCP/IP | | |
| 機能名 | WRITE | 現在のコネクションへデータを書き出す | |
| コマンド形式 | | $TW "data"¥n | |
| 引数 | data | 書き出すデータ | 最大1024バイト、$エンコードされていること |
| 応答値 | 【正常時】 | $TW=OK nbytes¥n | |
| | nbytes | 書き出したデータのバイト数（デコード後の生データのサイズ） | 最大1024バイト |
| | 【エラー時】 | $TW=NG errno ..¥n | |
| | errno | 301：引数指定エラー | |
| 前提条件 | ① | TCP/IPコネクションが確立されていること | |
| 補足事項 | ① | dataとして指定できるデータは$エスケープシーケンスにてエンコードされている必要がある。 | |
| | ② | dataとして指定できるデータは、エンコード前の生データのサイズが1024バイト以下であること。 | |

# 3．TCP/IP CONNECT

## ■コネクション接続

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | TCP/IP | | |
| 機能名 | CONNECT | TCP/IPコネクションを接続する | |
| コマンド形式 | | $TC host_or_ip port¥n | |
| 引数 | host_or_ip | 接続するホスト名またはIPアドレス | |
| | port | 接続するポート番号 | |
| 応答値 | 【正常時】 | $TC=OK¥n | |
| | 【エラー時】 | $TC=NG errno ..¥n | |
| | errno | 601：引数がおかしい | |
| | | 603：すでに接続済み | |
| | | 604：コネクション失敗 | |
| | | 605：内部エラー(Open) | |
| | | 606：内部エラー(Create) | |
| | | 607：内部エラー(SockPort) | |
| | | 609：タイムアウトエラー | |
| 前提条件 | ① | TCP/IPコネクションが確立されていないこと | |
| 補足事項 | ① | TCP/IPコネクションは一度に一つだけ使用できる。コネクションはWeb機能とは独立している。 | |

GET / HTTP/1.1
Accept: image/gif, image/jpeg, */*
Accept-Language: ja Accept-Encoding: gzip, deflate User-Agent: Mozilla/4.0 (Compatible; MSIE 6.0; Windows NT 5.1;) Host: www.xxx.zzz Connection: Keep-Alive

# 4．TCP/IP DISCONNECT

■コネクション切断

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | TCP/IP | | |
| 機能名 | DISCONNECT | 現在のTCP/IPコネクションを切断する | |
| コマンド形式 | | $TD¥n | |
| 引数 | | | |
| 応答値 | 【正常時】 | $TD=OK¥n | |
| | 【エラー時】 | $TD=NG errno ..¥n | |
| | errno | 614：内部エラー(Close) | |
| | | 615：接続されていない | |
| | | 616：内部エラー(Shutdown) | |
| 前提条件 | ① | TCP/IPコネクションが確立されていること | |
| | ② | read中あるいはwrite中ではないこと | |
| 補足事項 | | | |

# 5．TCP/IP STATUS

## ■コネクション状態の取得および状態設定

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | TCP/IP | | |
| 機能名 | STATUS | 現在の状態を取得する、指定した状態に設定する | |
| コマンド形式 | 取得 | $TS¥n | 現在の状態を取得する |
| | 設定 | $TS status¥n | 状態を強制的に設定する |
| 引数 | status | 0：CLOSED（接続なし） | |
| | | 1：DISCONNECTING | |
| | | 2：DISCONNECTED（接続待ち） | |
| | | 3：CONNECTING | |
| | | 4：CONNECTED（送受信待ち） | |
| | | 5：READING | |
| | | 6：WRITING | |
| 応答値 | 【正常時】 | $TS=OK status¥n | |
| | status | 上記参照 | |
| | 【エラー時】 | $TS=NG errno ..¥n | |
| | errno | 641：引数がおかしい | |
| 前提条件 | | | |
| 補足事項 | ① | 状態の設定は、問題を引き起こす可能性があるため使用しないこと。 | |
| | | （状態を変更しても、自動的に接続・切断等が実行される訳ではない） | |

# 6．TCP/IP GETSOCKNAME

■コネクション状態のIPアドレスとポート番号取得

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | TCP/IP | | |
| 機能名 | Get Sockname | 自分のIPアドレスおよびポート番号を取得する | |
| コマンド形式 | 取得 | $TN¥n | |
| 引数 | | | |
| 応答値 | 【正常時】 | $TN=OK ipAddr portNo¥n | |
| | ipAddr | 自分のIPアドレス(IP v4のみサポート) | |
| | portNo | 自分のポート番号 | |
| | 【エラー時】 | $TN=NG errno ..¥n | |
| | errno | 662：接続していない | |
| | | 661：内部エラー | |
| 前提条件 | ① | 相手に接続していること | |
| 補足事項 | ① | | |

# 7．TCP/IP 利用サンプルプログラム ①

## ■事例：TCP/IP関連一覧のコマンドを使ったサンプルプログラム

■Arduinoサンプルプログラム

```
#include <SoftwareSerial.h>
SoftwareSerial iemSerial(4,5);
void setup(){
 pinMode(7,OUTPUT);
 digitalWrite(7,LOW); delay(100); digitalWrite(7,HIGH);
 Serial.begin(9600);  iemSerial.begin(9600);
 Serial.println("Ready...");
 while(true) {
  unsigned long tim=millis();
  while(!iemSerial.available() && (millis() - tim)<30000);
  String st=iemSerial.readStringUntil('¥n');
  Serial.println(st);
 if( st.indexOf("gw3g")>0 ) { break;}
  else if (!((millis()-tim)<300000)) {
    Serial.println("connect error"); while(1);
  }
 }
 Serial.println("Start...");
 Serial.println("begin TCP/IP sample");delay(100);
 while(!tcpprintln("$TC www.tabrain.jp 80"));
 tcpprintln("$TW ¥"GET / HTTP/1.0$r$n¥"");
 tcpprintln("$TW ¥"HOST: www.tabrain.jp$r$n$r$n¥"");
 tcpprintln("$TR 200");
 tcpprintln("$TD");
 Serial.print("end");
}
void loop() {}
```

```
boolean tcpprintln(String ttc) {
 String rts="";
 uint32_t tm=millis();
 iemSerial.println(ttc);
 do{
   while(!iemSerial.available() && (millis()-tm<30000));
   rts=iemSerial.readStringUntil('¥n');
   Serial.println( rts );
 } while(!(rts.indexOf("$T")>=0));
 char ch;
 do{
     ch=iemSerial.read();
     if(0x20<ch && ch<0x80 ) Serial.print(ch);
  } while(iemSerial.available());
 return (rts.indexOf("=OK")>0);
}
```

# 7．TCP/IP 利用サンプルプログラム ②

## ■事例：TCP/IP関連一覧のコマンドを使ったサンプルプログラムの出力結果

■シリアルモニタ画面（正常時応答）

```
Ready...
ホ
Hello, I'm gw3g(Ver 2.0)
Start...
begin TCP/IP sample
>TC=IN PROGRESS TS_GetHostByName()

$TC=OK

$TW=OK 16

$TW=OK 24

>TR=ACCEPTED 200 bytes
$TR=OK 200
HTTP/1.1 200 OK
Date: Su
,13Sep2015 0tﾙﾑ◆ll
Server: Apache/2.2.23 (U
ix) mod_ssl/2.2.23 Ope
SSL/1.0.1m
Last-Modified: Mo
, 10 Aug 2015 07:28:44 GMT
ETag: "143577b-f884-51cefef82de24"
Acce
$TD=OK

end
```

読込みバッファの出力結果
（ここでは256バイト出力）


# 6．Profile関連

# 1．PROFILE SET ①

■SIMカードのプロファイル番号の設定

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | PROFILE | | |
| 機能名 | SET | デフォルトのプロファイル番号を設定する | |
| コマンド形式 | | $PS profileNum¥n | |
| 引数 | profileNum | プロファイル番号(1～16) | |
| 応答値 | 【正常時】 | $PS=OK¥n | |
| | 【エラー時】 | $PS=NG errno¥n | |
| | errno | 211：引数エラー | |
| 前提条件 | | | |
| 補足事項 | ① | あらかじめIEMに登録されているプロファイル情報については、下記のサイトを参照のこと：<br>http://a3gs.wiki.fc2.com/wiki/SIMカード情報 | |

# 1．PROFILE SET ②

■SIMカードのプロファイル番号の設定

■Arduinoサンプルプログラム

```
#include <SoftwareSerial.h>
SoftwareSerial iemSerial(4,5);
void setup(){
  Serial.begin(9600);
  iemSerial.begin(9600);
  delay(35000);
  Serial.println("begin Profaile Set");
  iemSerial.print("$PS 2¥n");
  while(!iemSerial.available());
  String rt = iemSerial.readStringUntil('¥n');
  Serial.print(rt);
  Serial.print("¥r¥nend");
}
void loop() {}
```

■シリアルモニタ画面（正常終了の場合）

```
begin Profaile Set
$PS=OK
end
```

# 2．PROFILE READ ①

■SIMカードのプロファイル情報の読出し

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | PROFILE | | |
| 機能名 | READ | 指定したプロファイル情報を読み出す | |
| コマンド形式 | 取得 | $PR¥n | デフォルトのプロファイル番号を取得 |
| | 読み出し | $PR profileNum¥n | 指定したプロファイル情報を読み出し |
| 引数 | pfileNum | プロファイル番号(1～16) | |
| 応答値 | 【正常時】 | $PR=OK profileNum¥n | 引数なしの時 |
| | profileNum | デフォルトのプロファイル番号 | |
| | 【正常時】 | $PR=OK apn auth authtype pwd usr dns dns1 dns2¥n | 引数指定の時 |
| | apn | APN名 | |
| | auth | 認証情報 | |
| | authtype | 認証タイプ(AUTH_NONE、AUTH_PAP または AUTH_CHAP のいずれか) | |
| | pwd | パスワード | 設定なしの場合は「-」を出力 |
| | usr | ユーザ名 | 同上 |
| | dns | DNS情報（dns1とdns2をセミコロンで連結した値） | |
| | dns1 | プライマリDNS | 同上 |
| | dns2 | セカンダリDNS | 同上 |
| | 【エラー時】 | $PR=NG errno¥n | |
| | errno | 201：引数エラー | |
| 前提条件 | | | |
| 補足事項 | | | |

# 2．PROFILE READ ②

■事例：SIMカードのプロファイル情報出力のプログラム

■Arduinoサンプルプログラム

```
#include <SoftwareSerial.h>
SoftwareSerial iemSerial(4,5);
void setup(){
  Serial.begin(9600);
  iemSerial.begin(9600);
  delay(35000);
  Serial.println("begin Profile Read");
  iemSerial.print("$PR¥n");
  while(!iemSerial.available());
  String rt = iemSerial.readStringUntil('¥n');
  Serial.print(rt);
  Serial.print("¥r¥nend");
}
void loop() {}
```

■シリアルモニタ画面（"$PR¥n" の場合）

```
begin Profile Read
$PR=OK 2
end
```

■シリアルモニタ画面（"$PR 2¥n" の場合）

```
begin Profaile Read
$PR=OK iijmio.jp AUTH_PAP;iij;mio@iij AUTH_PAP iij mio@iij 0.0.0.0;0.0.0.0 - -
end
```

■シリアルモニタ画面（ "$PR 5¥n"の場合）

```
begin Profaile Read
$PR=OK so-net.jp AUTH_PAP;nuro;nuro AUTH_PAP nuro nuro 0.0.0.0;0.0.0.0 - -
end
```

# 3．PROFILE RESET ①

■SIMカードのプロファイル情報のクリア

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | PROFILE | | |
| 機能名 | RESET | 指定したプロファイル情報をクリアする | |
| コマンド形式 | | $PE profileNum¥n | |
| 引数 | profileNum | プロファイル番号(1～16) | |
| 応答値 | 【正常時】 | $PE=OK¥n | |
| | 【エラー時】 | $PE=NG errno¥n | |
| | errno | 231：引数エラー | |
| | | 232：内部エラー（リセットエラー） | |
| 前提条件 | | | |
| 補足事項 | ① | 現在の実装では、常にエラー(232)となるが、リセット処理は正常である。 | |
| | ② | 本機能は工場出荷時の設定に戻す機能であるため、指定プロファイルにAPN情報がプリセットされている場合は、プリセットされているAPN情報は残る。 | |

# 3．PROFILE RESET ②

■事例：SIMカードのプロファイル情報のクリアプログラム

■Arduinoサンプルプログラム

```
#include <SoftwareSerial.h>
SoftwareSerial iemSerial(4,5);
void setup(){
  Serial.begin(9600);
  iemSerial.begin(9600);
  delay(35000);
  Serial.println("begin Profile Reset");
  iemSerial.print("$PE 7¥n");
  while(!iemSerial.available());
  String rt = iemSerial.readStringUntil('¥n');
  Serial.println(rt);
  Serial.println("end");
}
void loop() {}
```

■シリアルモニタ画面（現時点通常の応答）

```
begin Profile Reset
$PE=NG 232
end
```


# 7．Storage関連

# 1．STORAGE WRITE ①

## ■IEM(通信）モジュールのストレージへのデータ書込み

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | Storage | | |
| 機能名 | WRITE | ストレージへのデータ書込み | |
| コマンド形式 | | $RW no "data"¥n | |
| 引数 | no | ストレージ番号（1～10） | |
| | data | ストレージへ書き込むデータ($エスケープされた文字列) | $エスケープ前で最大1023バイト |
| 応答値 | 【正常時】 | $RW=OK¥n | |
| | | | |
| | 【エラー時】 | $RW=NG errno ..¥n | |
| | errno | 711：引数指定エラー | |
| | | 712：内部エラー | |
| 前提条件 | | gw3g R2.0以降のみで利用できる | |
| 補足事項 | | | |

# 1．STORAGE WRITE ②

■事例：通信モジュール・ストレージへのデータ書出し

■Arduinoサンプルプログラム

```
#include <SoftwareSerial.h>
SoftwareSerial iemSerial(4,5);
void setup(){
  Serial.begin(9600);
  iemSerial.begin(9600);
  delay(30000);
  Serial.println("begin Strage Read");
  iemSerial.print("$RW 1 ¥"Test Strage Write¥"¥n");
  while(!iemSerial.available());
  String rt = iemSerial.readStringUntil('¥n');
  Serial.println(rt);
  rt = iemSerial.readStringUntil('¥n');
  Serial.println(rt);
  Serial.println("end");
}
void loop() {}
```

ストレージ書出しする文字列

■シリアルモニタ画面（正常の場合）

```
begin Strage Write
$RW=OK 17
end
```

# 2．STORAGE READ ①

■IEM(通信）モジュール・ストレージからのメモリ読込み

| 項目 | 値など | 説明 | 補足 |
|---|---|---|---|
| 機能分類 | Storage | | |
| 機能名 | READ | ストレージからのデータを読み込み | |
| コマンド形式 | | $RR no¥n | |
| 引数 | no | ストレージ番号（1～10） | |
| 応答値 | 【正常時】 | $RR=OK nbytes¥ndata¥n | |
| | data | 読み出したデータ | 最大1023バイト、バイナリデータも取扱可 |
| | 【エラー時】 | $RR=NG errno ..¥n | |
| | errno | 701：引数指定エラー | |
| | | 702：データ無し（指定されたストレージにはデータがない） | |
| 前提条件 | ① | gw3g R2.0以降のみで利用できる | |
| 補足事項 | | | |

# 2．STORAGE READ ②

■事例：IEM(通信）モジュール・ストレージからのメモリ読込みプログラム

■Arduinoサンプルプログラム

```
#include <SoftwareSerial.h>
SoftwareSerial iemSerial(4,5);
void setup(){
  Serial.begin(9600);
  iemSerial.begin(9600);
  delay(30000);
  Serial.println("begin Strage Read");
  iemSerial.print("$RR 1¥n");
  while(!iemSerial.available());
  String rt =
iemSerial.readStringUntil('¥n');
  Serial.println(rt);
  rt = iemSerial.readStringUntil('¥n');
  Serial.println(rt);
  Serial.println("end");
}
void loop() {}
```

■シリアルモニタ画面（正常の場合）

```
begin Strage Read
$RR=OK 17
Test Strage Write
end
```

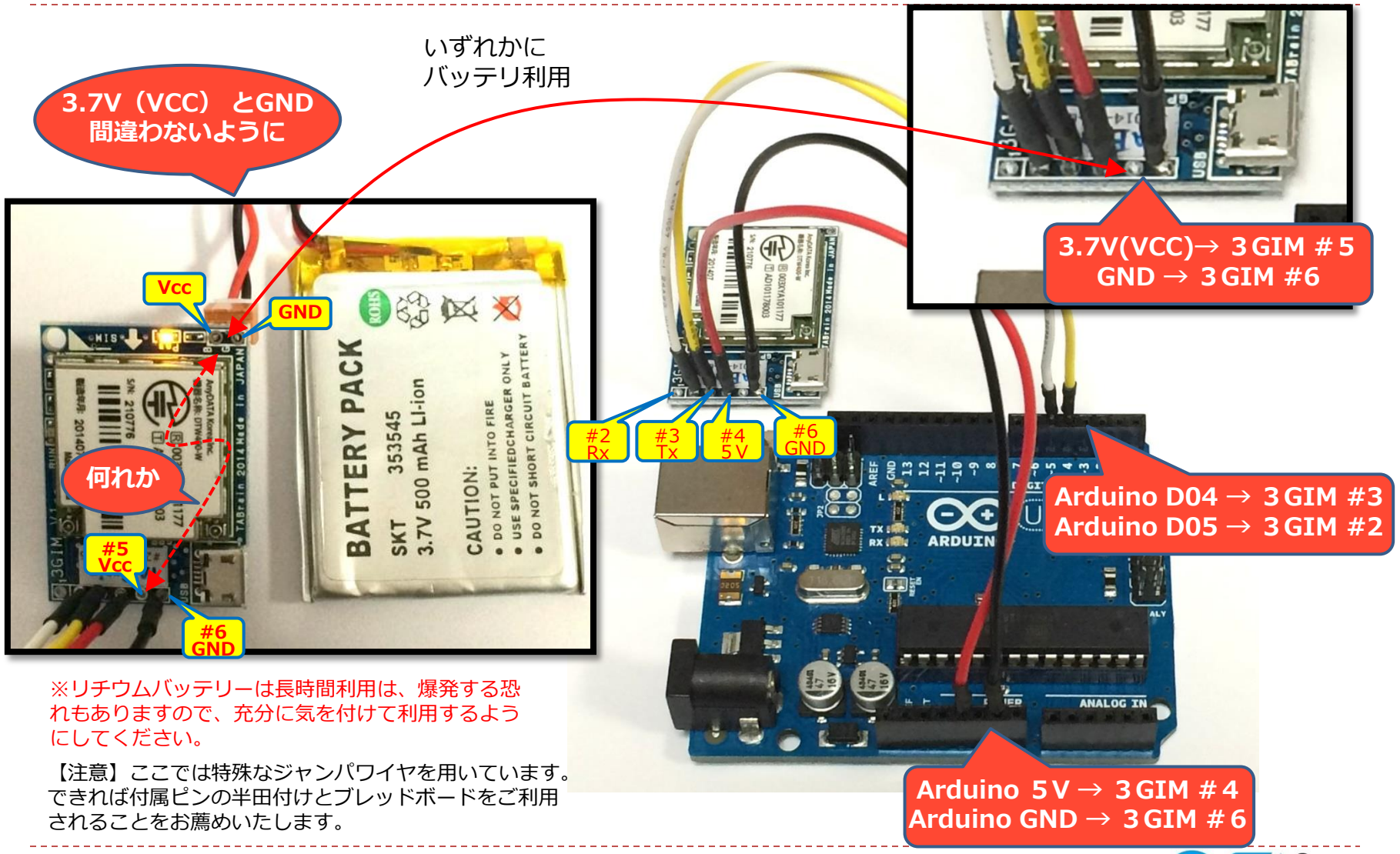
もくじ




# 第3章
# 応用事例プログラミング

# １．Arduinoでのシリアルモニタ操作

# 1．Arduino UNO との接続例




※リチウムバッテリーは長時間利用は、爆発する恐れもありますので、充分に気を付けて利用するようにしてください。

【注意】ここでは特殊なジャンパワイヤを用いています。できれば付属ピンの半田付けとブレッドボードをご利用されることをお薦めいたします。

# 2．Arduino上での簡単な利用例

- 3Gシールドで紹介しているサンプルスケッチは、すべて3GIMで動かすことができます。
  以下が3Gシールド用の環境とArduinoサンプルスケッチです。
  解説書： IEM製品版3Gシールドライブラリ仕様書（Ver2.0）
  ※注意点として、デフォルトの通信速度が異なります※
  3Gシールド V1.2（4800bps）に対して3GIM（9600bps）です。
  ＜3GシールドV2.0では9800bpsとなっています＞
  a3gsa.h の中の 「a3gsBAUDRATE」の値 を4800から9600へ変更してください。

**【デフォルトの通信速度が異なる理由】**
3GシールドはArduino UNOをターゲットとしているためSoftwareSerialで文字化けしない4800bpsという低い速度をデフォルトの設定としています。
一方、3GIMはArduino以外のマイコンでも利用できることを特長としていまして、広く一般的に利用されている9600bpsをデフォルトの設定として採用しています。

■必要な部品：
① スルーホール用テストワイヤ
② 3.7V リチウムイオン電池　（または VCC#5入力による電源【補足資料2】参照）
③ SIMカード（利用できるSIMはWikiページ http://a3gs.wiki.fc2.com を参照）

# 3．Arduinoシリアルモニタ画面操作スケッチ

- Arduinoと３GIMを接続し、シリアルモニタ画面上でコマンド入力して、その結果を見てみることにしてみましょう。
- Arduinoのスケッチは以下のとおりです。

■ シリアルモニタ画面での３GIM入出力プログラム例

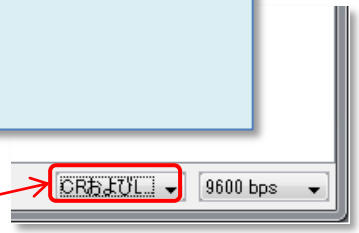
```
#include <SoftwareSerial.h>
SoftwareSerial  iemSerial(4, 5);
const unsigned long baudrate = 9600;

void setup() {
  Serial.begin(baudrate);
  iemSerial.begin(baudrate);
  pinMode(7,OUTPUT);
  digitalWrite(7,HIGH);
  Serial.println("Ready.");
}

void loop() {
  if (iemSerial.available() > 0) {
    char c = iemSerial.read();
    Serial.print(c);
  }

  if (Serial.available() > 0) {
    char c = Serial.read();
    Serial.print(c);    // Echo back
    iemSerial.print(c);
  }
}
```

本サンプルスケッチは、
シリアルモニタ画面で、コマンドをキー入力することで、応答（レスポンス）を表示確認できるもので、マニュアル操作でのコマンド/レスポンスが即座に見ることができます。

■ シリアルモニタ画面での操作例

```
Ready.
$YV
$YV=OK 2.0
$YI
$YI=OK 356423042110000
$YR
$YR=OK -71
$YT
$YT=OK 2015/03/31 17:55:11
$LG MSBASED
>LG=START 4
$LG=OK 35.64189613 139.6041995
$WG http://tabrain.jp/demo/httpGET_test.txt
>W*=STARTING
>W*=GETHOSTBYNAME
>W*=CONNECT
>W*=SENDREQUEST
>W*=READRESPONSE
>WG=CONTENT_TYPE text/plain
>WG=READ(44Byte)
$WG=OK 44
Tabrain Web site
Complete access from 3GIM
```

赤字：入力
青字：出力



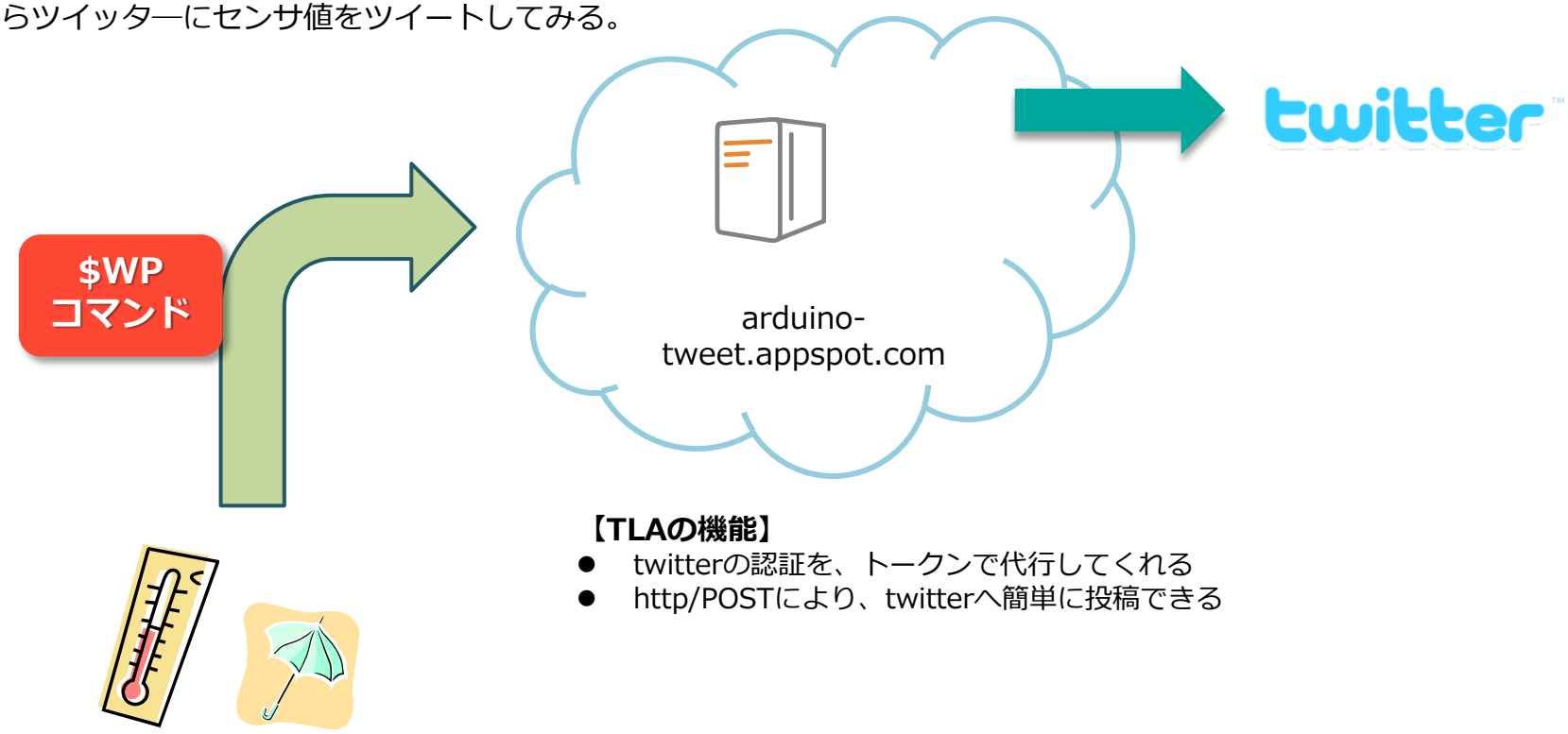

補足：Arduino IDEのシリアルモニタ画面の改行モードは「CRおよびLF」に設定のこと

# ２．3GIMでのツイッター連携使用例（Arduinoの事例）

# 1．３GIMを使ったツイッター連携の利用イメージ

**■TLAの利用**

Tweet Library for Arduino【TLA】を利用し、３GIMからツイッターにセンサ値をツイートしてみる。




**【TLAの機能】**

- twitterの認証を、トークンで代行してくれる
- http/POSTにより、twitterへ簡単に投稿できる

# 2．TLA利用手順①

①ブラウザで、下記のサイトにアクセスする
http://arduino-tweet.appspot.com/

# 2．TLA利用手順②

② ツイッターのアカウント情報を入力する
（既にツイッターID登録済の場合には次に進んでください）

# 3．TLA利用手順③

③ トークンを記録しておく（コピー＆ペースト）

# 4．３GIMでのツイッター連携使用例

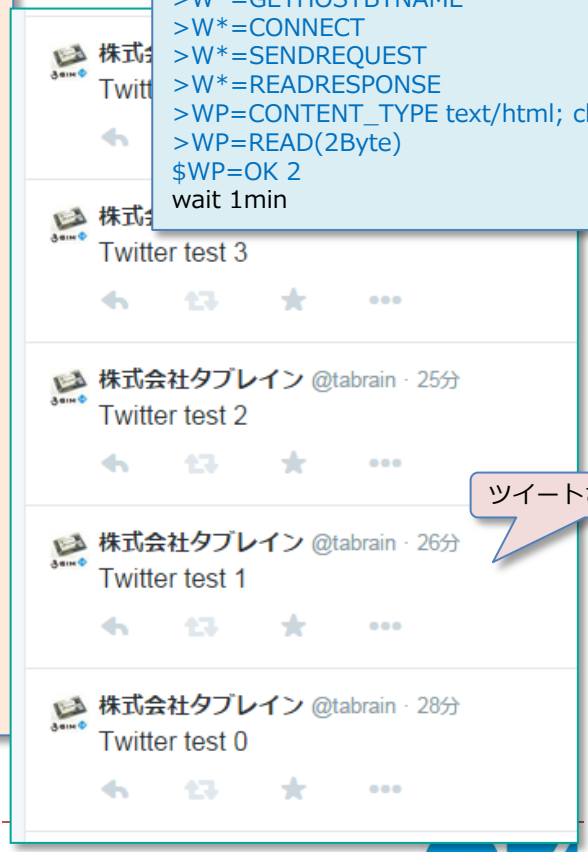
■事例：ツイッターにメッセージアップのプログラム

```
#include <SoftwareSerial.h>
SoftwareSerial iemSerial(4,5);
const char *server = "http://arduino-tweet.appspot.com:80/update";
const char *token = "YOUR_TOKEN";

void setup() {
  Serial.begin(9600);
  iemSerial.begin(9600);
  Serial.print(" Plsese waite");
  delay(35000);
 Serial.println("Ready");
}

void loop() {
  char msgBuff[300];
  char msg[]="Twitter test";
  static int no = 0;//  番号のカウントアップ
  sprintf(msgBuff,"$WP %s ¥"token=%s&status=%s %d¥"¥n",server,token,msg,no++);
  Serial.println(msgBuff);
  iemSerial.println(msgBuff);
  String rt;
  do{
     while(!iemSerial.available());
     rt = iemSerial.readStringUntil('¥n');
     Serial.println(rt);
  } while(!rt.startsWith("$WP"));
  Serial.println("wait 1min");
  delay(60000);//   ツイートは1分毎とする（制限事項）
}
```

ツイートのトークン

■実行モニタ画面（正常時）

```
 Plsese waite
Ready
$WP ************

>W*=STARTING
>W*=GETHOSTBYNAME
>W*=CONNECT
>W*=SENDREQUEST
>W*=READRESPONSE
>WP=CONTENT_TYPE text/html; charset=utf-8
>WP=READ(2Byte)
$WP=OK 2
wait 1min
```




ツイートされた画面

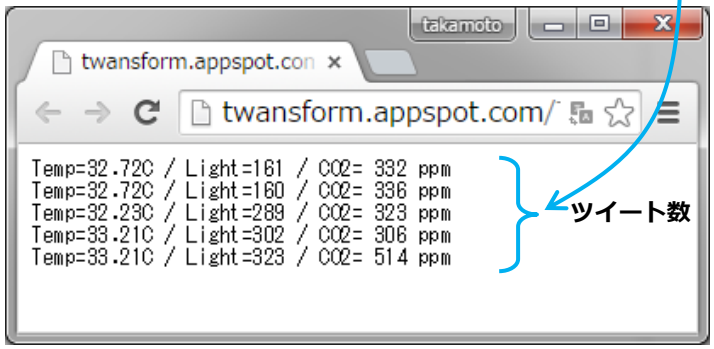
# 5．ツイートの読込み①

■ツイートされた内容（値や文章）をクライアント側（３G端末側）に読込むサンプル

ツイートされた内容を読むには、以下のアクセスで可能

http://twansform.appspot.com/**ツイッター名**/text/5

<ブラウザでキー入力すると以下のような表示が返る>

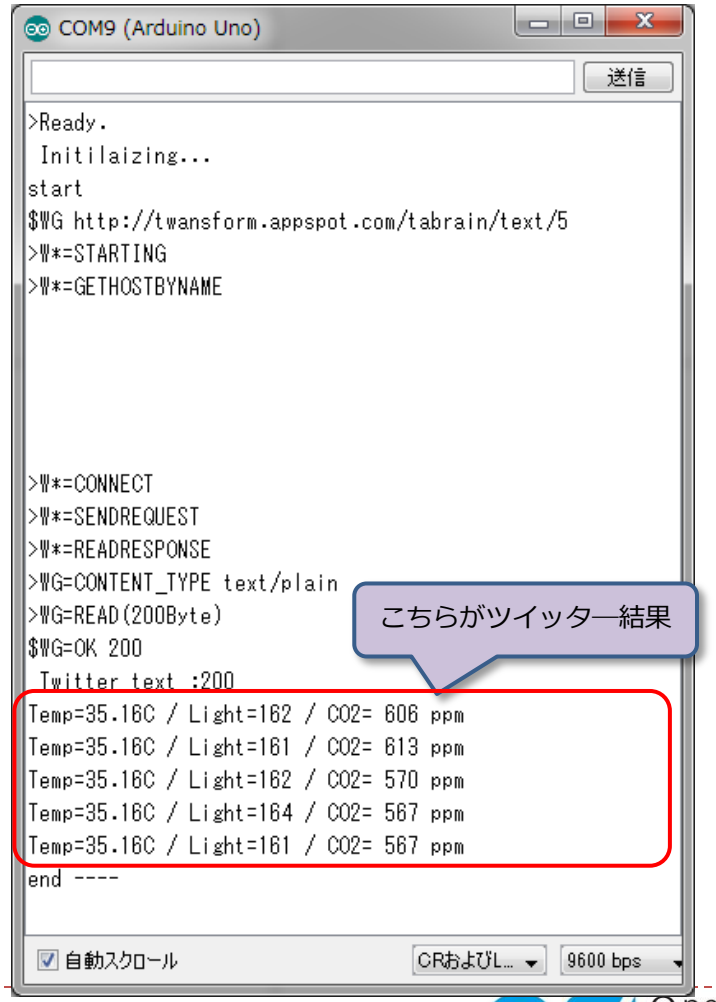



■今度は、３G通信を使ってツイートされた内容（値や文章）を読み込むには、以下のスケッチなどで行う。

$WG http://twansform.appspot.com/**ツイッター名**/text/**行数**

※ここで ツイッター名は、@で始まるツイッター名で、@を取り除いた後ろの名前。行数は、最新版から取得するツイッター数

■次頁のサンプルスケッチで実行した結果



# 5．ツイートの読込み②

■ツイートされたデータの読込み

（３GIMの電源信号（ピン番号＃１）は、Arduino側のD7ピンに接続）

```
#include <SoftwareSerial.h>
SoftwareSerial iemSerial(4,5);
SoftwareSerial D300(8,9);
const unsigned long baudrate = 9600;

#define LIMITTIME 35000 // ms (3G module start time)

String command  = "$WG http://twansform.appspot.com/ツイッタ名
/text/5";

//=============================================
void setup() {
  Serial.begin(baudrate);
  Serial.println(">Ready. ¥r¥n Initilaizing...");
  if( _3Gsetup() ) {
     Serial.println("start");
  } else {
     Serial.println(" Connect Error ... Stop");
     while(1);
  }
 while(! _3G_WG(command));
  Serial.println("end ----");
}
//=============================================
void loop () {}
```

成功するまで実行

■応用例

ツイートしたことで遠隔制御も可能

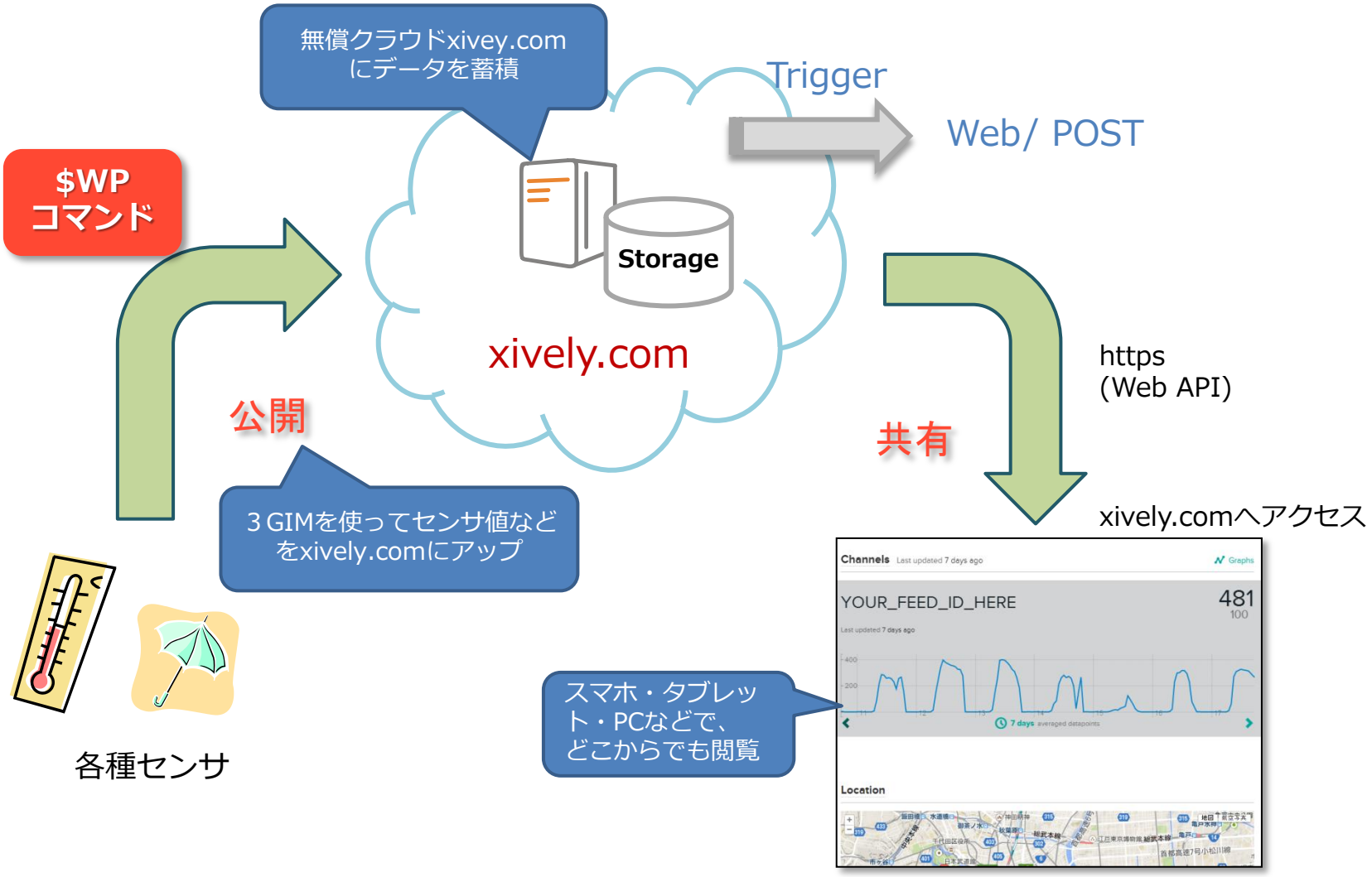
```
//============== 3G setup ========
boolean _3Gsetup() {
  pinMode(7,OUTPUT);
  digitalWrite(7,HIGH); delay(1000);// 3Gshield --> digitalWrie(7,LOW);
  digitalWrite(7,LOW); delay(100);  // 3G shield --> digitalWrite(7,HIGH);
  iemSerial.begin(baudrate);
 //------- 3G module begin & connect ---------
  String str;
  unsigned long tim = millis();
  do{  while(!iemSerial.isListening());
       str=iemSerial.readStringUntil('¥n');
  }while(!(str.indexOf("gw3g")>0) && (millis() - tim) <LIMITTIME);
  if( millis() -tim >= LIMITTIME) {
     return false;
  } else return true;
}
//============== $WG command ================
boolean _3G_WG(String command) {
  iemSerial.println(command);   Serial.println(command); // debug
  String rstr;
  unsigned long tim = millis();  // time set(ms)
  do{ while(!iemSerial.isListening());
       rstr=iemSerial.readStringUntil('¥n');
       Serial.println(rstr);  //debug print....
  }while(!(rstr.indexOf("$WG=")==0));// $WP return check
  if(rstr.indexOf("$WG=OK")==0) {
   rstr=rstr.substring(7);    int N=rstr.toInt();
    Serial.println(" Twitter text :" + rstr);
    for(int i=0; i<200; i++) {
      while(!iemSerial.available());
      char c=iemSerial.read();
      Serial.print(c);
    }
    return (true);
  }
  return(false);
}
```

３Gシールドの場合

こちらを一部追加

# ３．3GIMでのクラウド連携使用例①
# （xibely.com ＆ Arduinoの事例）

# 1．xively.comの利用イメージ



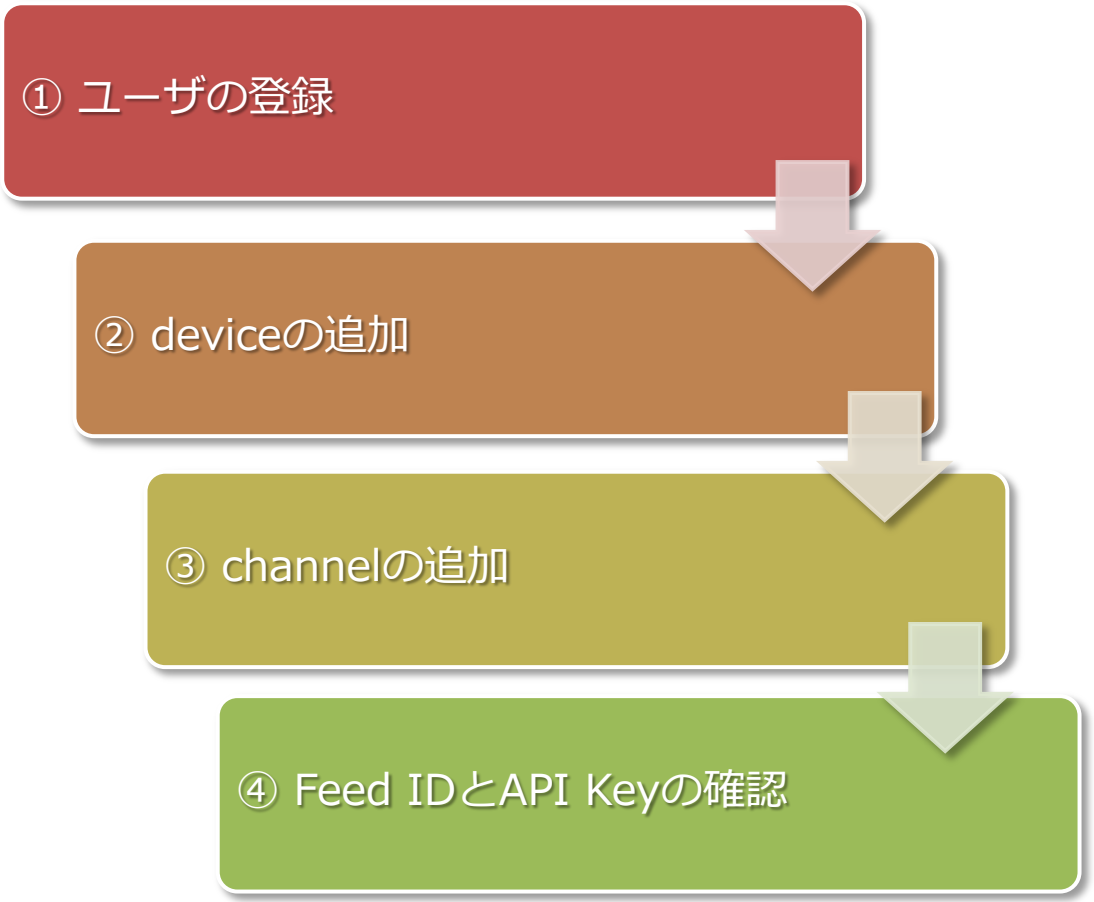
# 2．xivey.comの利用手順

xively.comは、実績が豊富な無償のクラウドで、日本でも広く利用されています。
まだ、英語版しかありませんが、グラフ表示やデータのアップやダウンロードだけの利用と考えると、問題なく簡単に使うことができます。

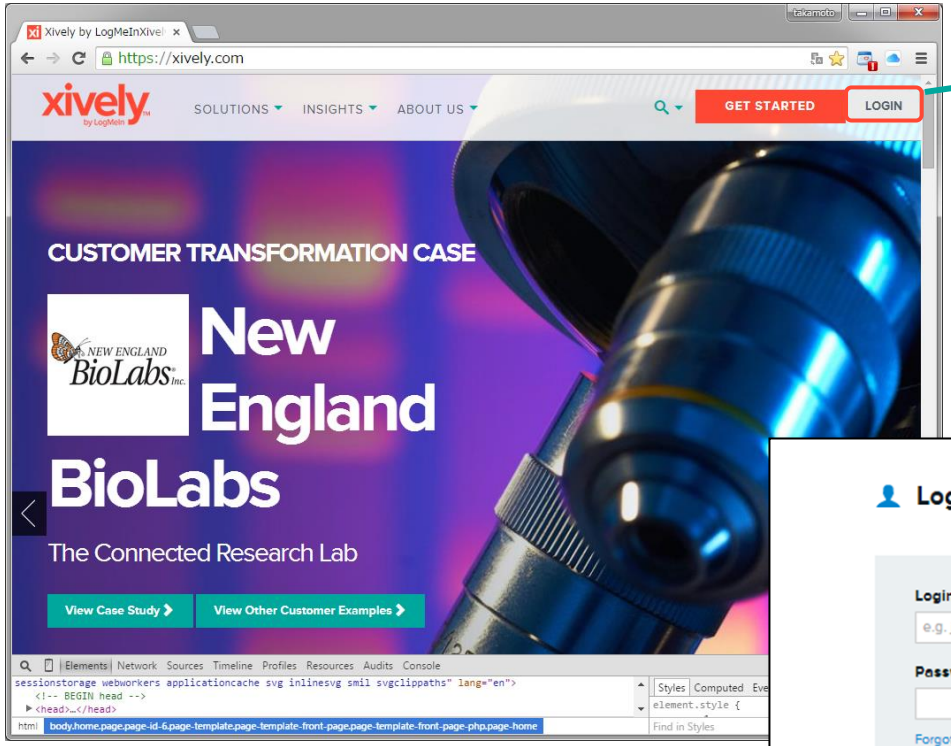
ここでは、本３GIMとセンサなどを使い、このxively.comにデータをアップしていくサンプルをご紹介します。

先ずは、xively.comでの①ユーザ登録が必要で、その後各設定（②deviceの追加と③channelの追加）を行い、それら設定された値（④Feed IDとAPI Keyの確認）を使うことで、プログラミングしていきます。

【利用に当たっての注意点】
xively.comでは、無償の範囲での利用は、制限があります。特にデータのアップは、1分間に数回程度でしかできません。
1秒毎とか頻繁にデータをアップしたりすると、利用できなくなることがあります。
充分に気を付けてプログラミングしてください。

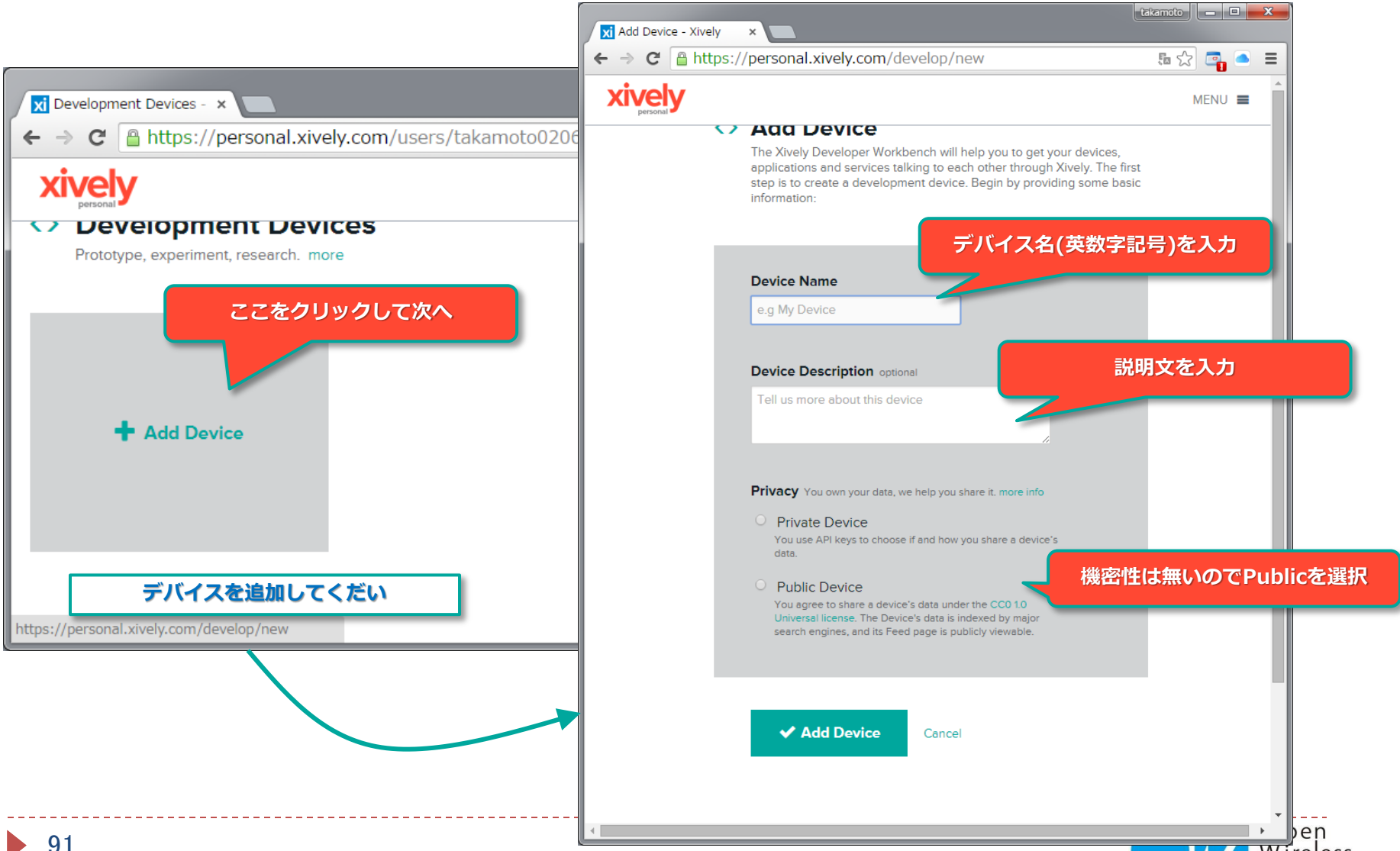


# 3．xively.com ①ユーザ登録

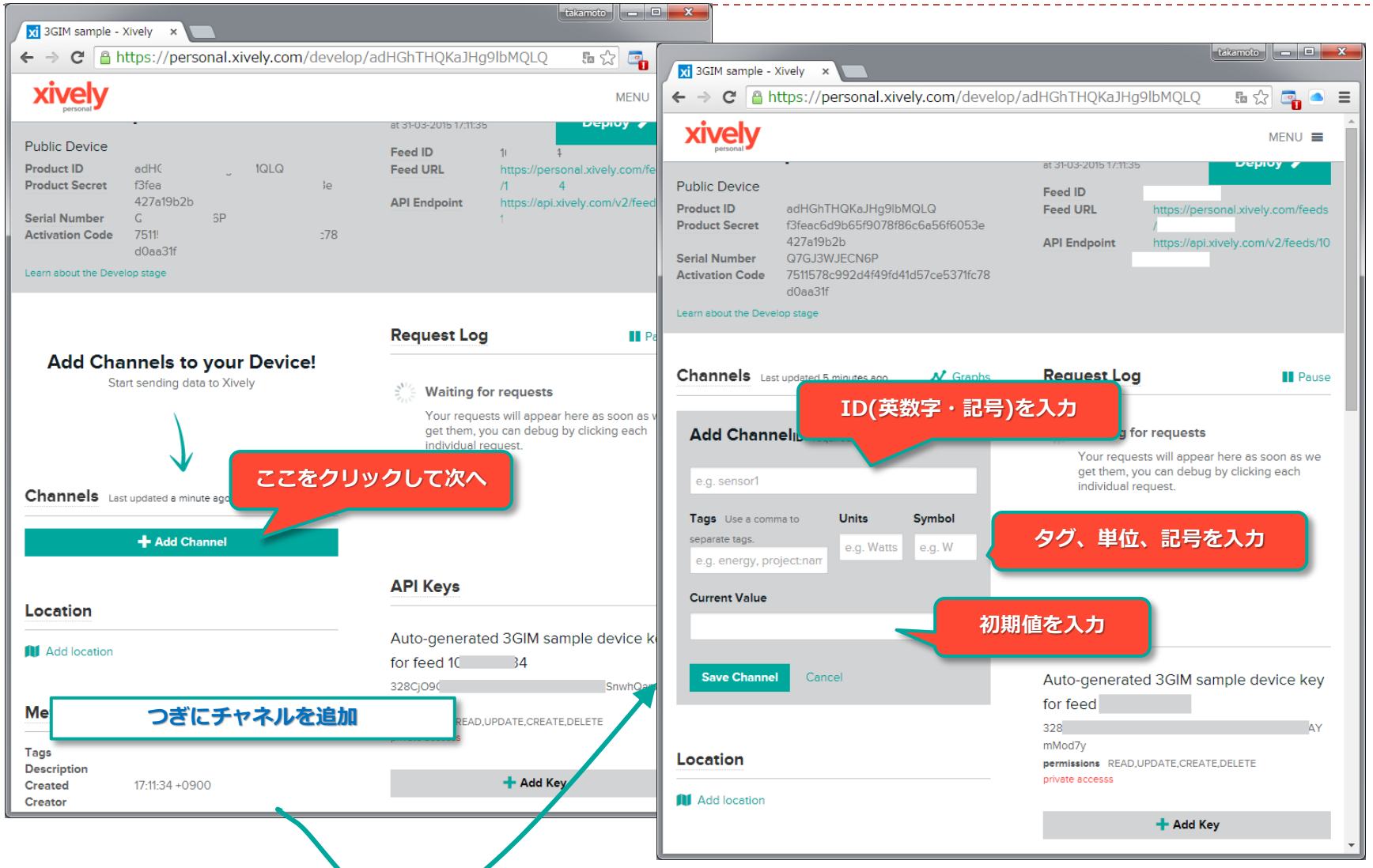



http://xively.com/ にアクセス

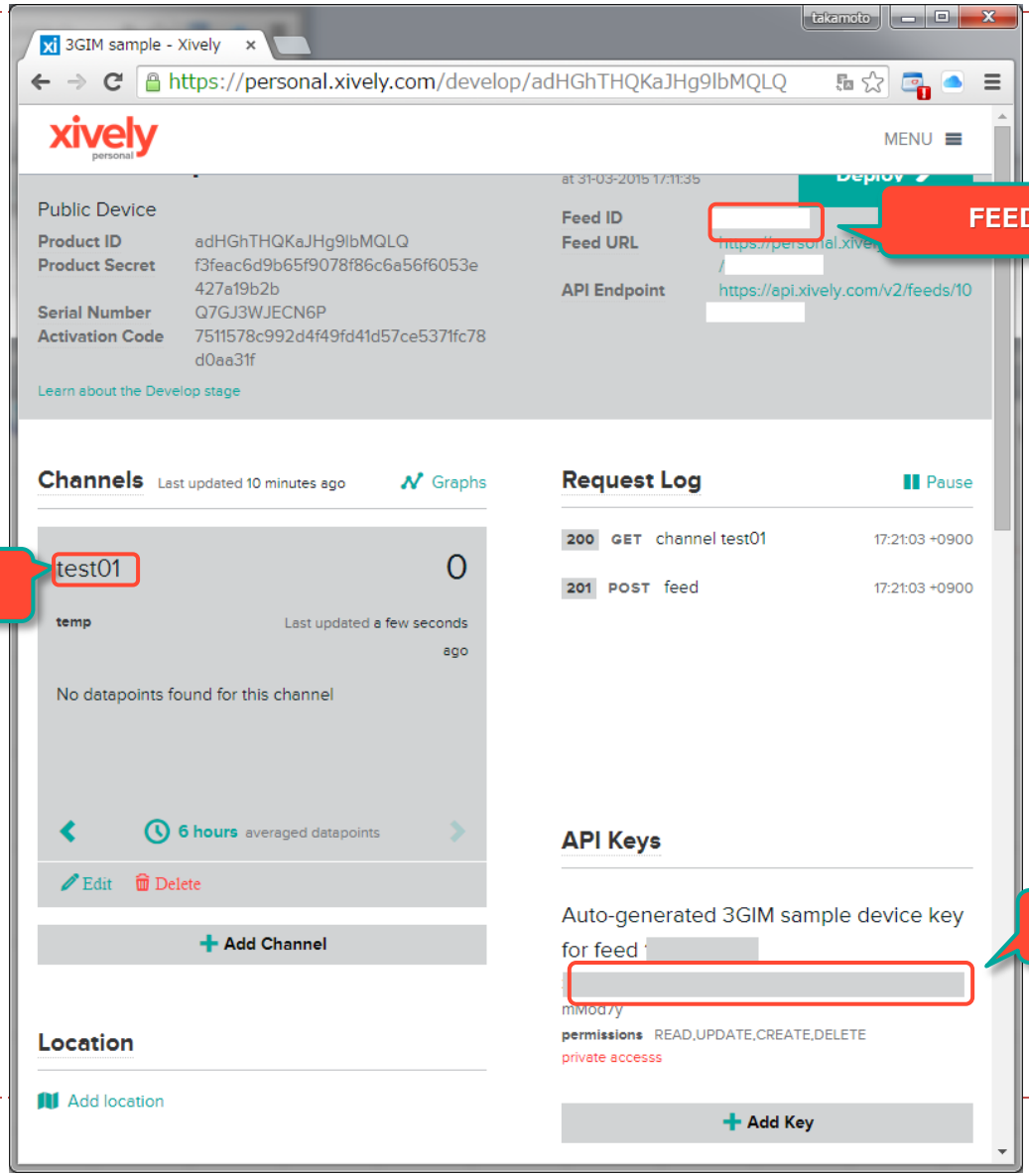
# 4．xively.com ②deviceの追加



# 5．xively.com ③chenellの追加



# 6．xively.com ④Feed IDとAPI Keyの確認



# 7．温度を測って定期的にxively.comへアップ

- **準備するもの**
  - Arduino UNO R3 など
  - 温度センサ（LM61BIZ）など
  - ブレッドボード
  - ジャンパ線（やわらかい線）
  - 3GIM（あらかじめピンヘッダを半田付けしておく）
  - マイクロSIMカード（3GIMで使えるもの）
  - 3.7Vリチウムポリマ電池（充電してあるもの）、または3.7V出力可能なDC電源

- **接続方法**
  - 3GIMにマイクロSIMを挿入して、ブレッドボードにピンヘッダを刺す。
  - #6(GND)を電源（リチウムポリマ電池）のGNDとArduinoのGNDに接続、
  - #5(VCC)を電源（リチウムポリマ電池）の「＋」に接続
  - #4(IOREF)をArduinoの5V、#3(TX)をArduinoの**D4**、
    #2(RX)をArduinoの**D5**、#1(PWR_ON)を**D7**に、それぞれジャンパ線で接続する。
  - 温度センサをブレッドボードに刺して、センサのGNDをArduinoのGND、VddをArduinoの5V、Voutを
    Arduinoの**A0**に、それぞれジャンパ線で接続する。

    （※ここで、**D4**,**D5**,**D7**および**A0**は、Arduino I/Oポート入出力番号）

# 8．温度を測って定期的にxively.comへアップ

- **サンプルスケッチ**

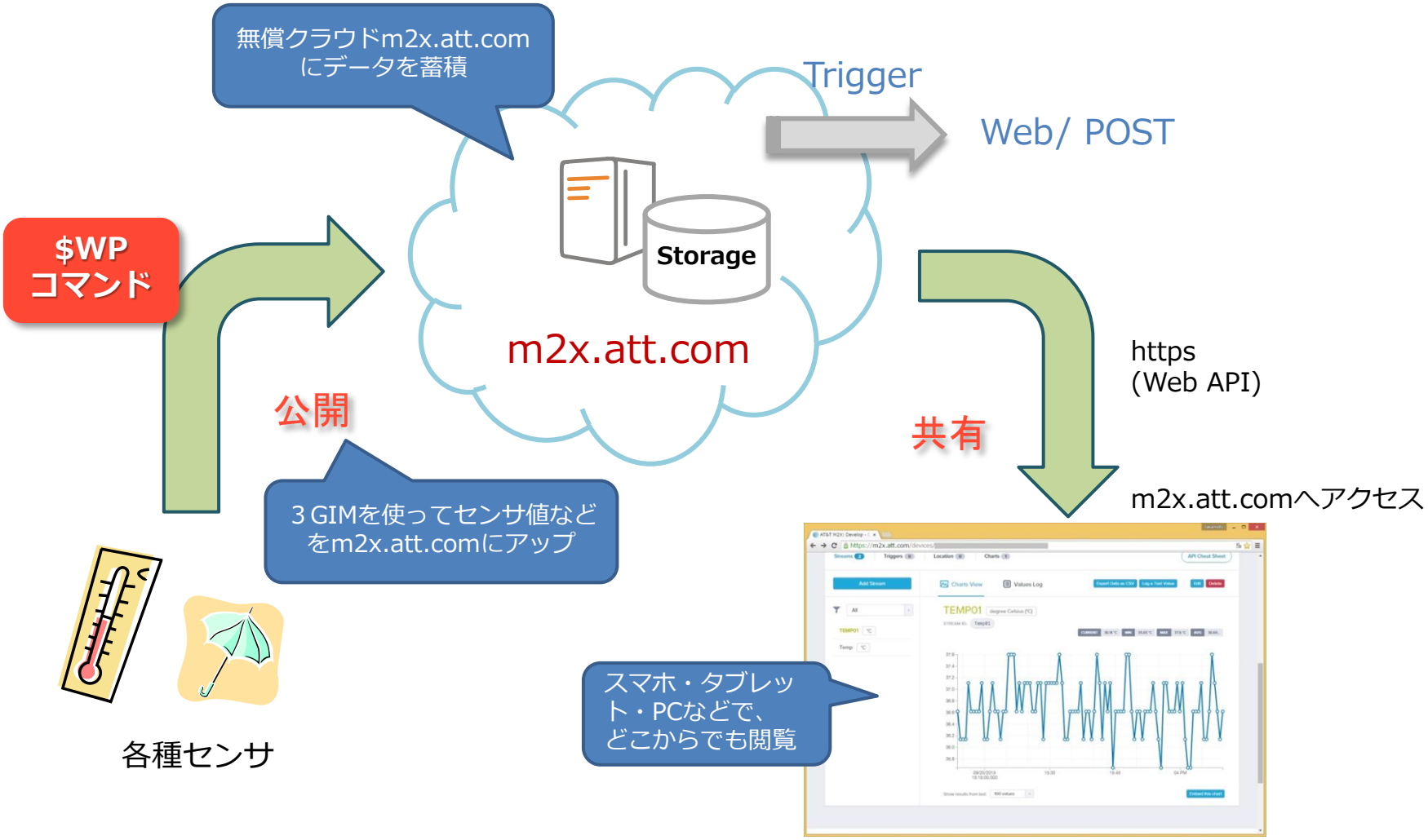
```
//  Sample sketch for 3GIM

#include <SoftwareSerial.h>

const int PowerPin = 7;  // D7
const int tmpPin    = 0;  // A0
const char *PostCmd = "$WP https://api.xively.com/v2/feeds/FEED_ID/datastreams/ CHANNEL_ID?_method=put ";
const char *Header  = "¥"X-ApiKey: API-KEY$r$nContent-Type: text/csv$r$n¥"";

// Global variables
uint32_t interval   = 60000;  // Interval time [mS]   1min
SoftwareSerial  iemSerial(4, 5);
char body[20];

// setup() -- set up device
void setup() {
    pinMode(PowerPin, OUTPUT);
    digitalWrite(PowerPin, LOW);    // 3GIM on
    iemSerial.begin(9600);
    delay(35000);    // wait for start up 3gim
}
void loop() {
    // Sense temperature
    int tX10 = getTemperature() * 10;
    // upload sensing data to the xively.com
    uploadToCloud(tX10);
    // sleep a while
    delay(interval);
}
```

赤文字の箇所は、実際のxively.comの登録内容に沿って修正すること！

```
float getTemperature() {
    int mV = analogRead(tmpPin) * 4.88;
    return ((float)(mV - 600) / 10.0);
}

void uploadToCloud(int temp) {
    // upload temperature
    iemSerial.print(PostCmd);
    sprintf(body, "¥"%d.%d¥" ", (temp / 10), abs(temp % 10));
    iemSerial.print(body);
    iemSerial.println(Header);
    iemSerial.flush();
}
```

# ４．3GIMでのクラウド連携使用例②
# （M2X ＆ Arduinoの事例）

# 1．M2X（AT&T IoTサービス）の利用イメージ

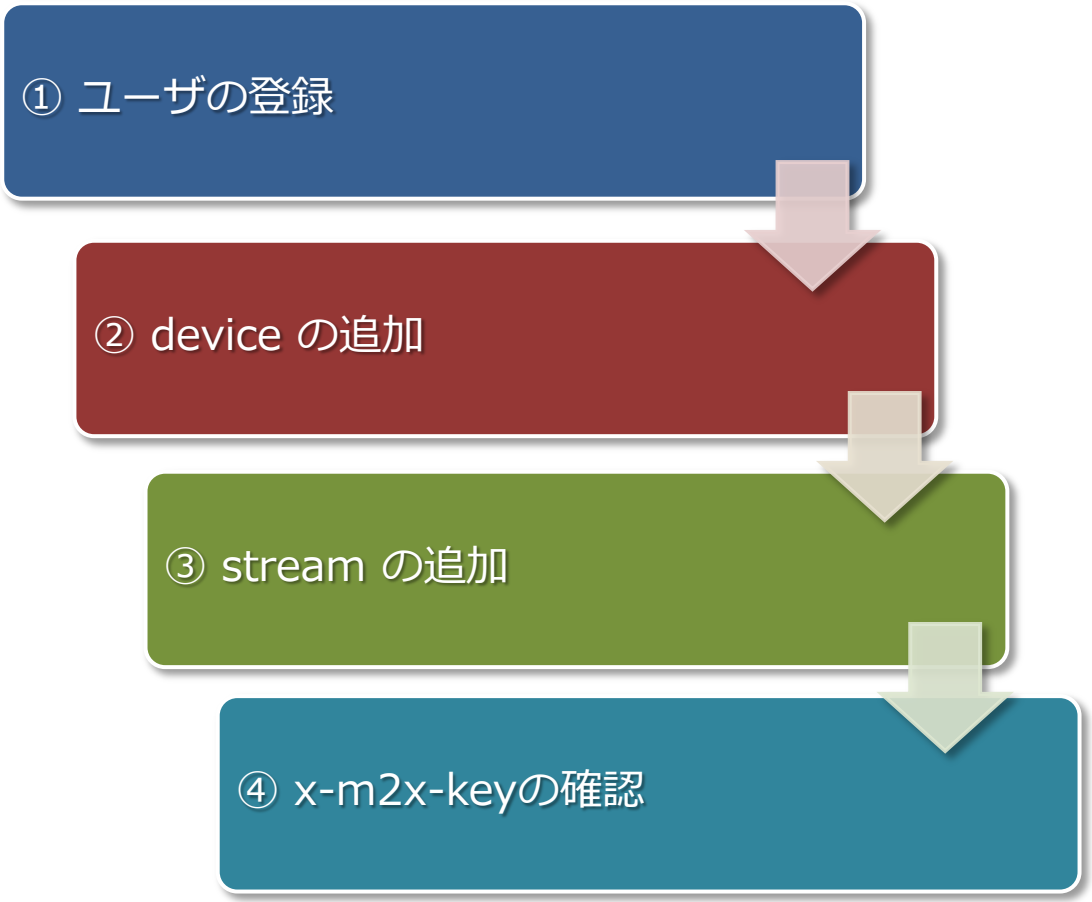


# 2． M2X（AT&T IoTサービス）の利用手順

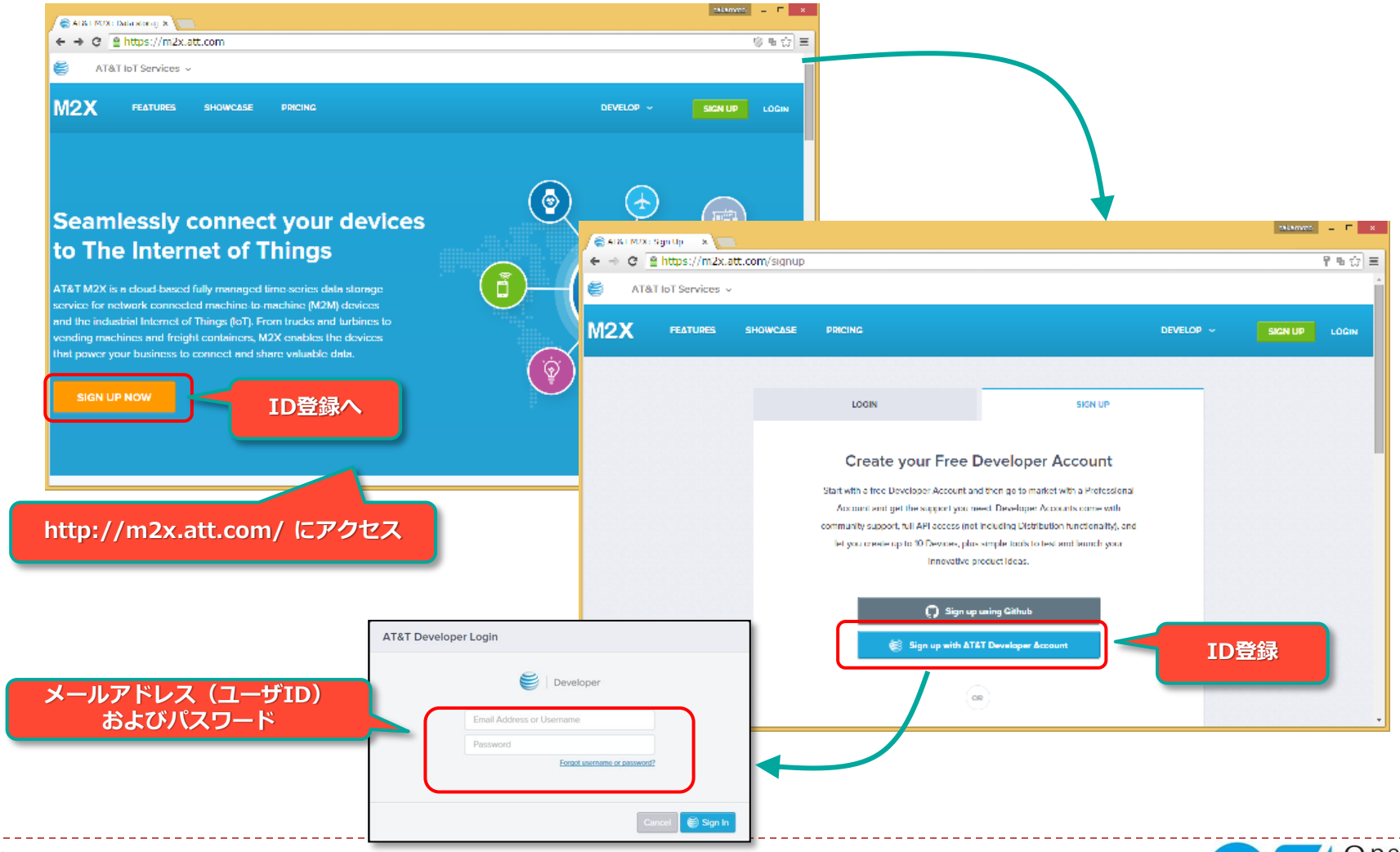
M2X（AT&T IoTサービス）は、2013年から米国通信事業最大手のAT&Tが、M2MおよびIoTビジネスに向けたサービスを開始したものです。
ArduinoやRasberryPi、Mbedなど、多くのオープンソースハードウェアで利用できる環境を提供したものとなっています。
フリーで使え、センサデータの蓄積・グラフ表示、データのダウンロードなどができるようになっています。
ただ、時間設定が、世界標準のみで行なっていて、日本時間での表示が現時点できないのが難点となっています。
ただ、登録の簡単さは
まだ、英語版しかありませんが、グラフ表示やデータのアップやダウンロードだけの利用と考えると、問題なく簡単に使うことができます。

ここでは、本３GIMとセンサなどを使い、このm2x.att.comにデータをアップしていくサンプルをご紹介します。

詳細な規約等は、m2x関連の公開情報等をご参照ください。

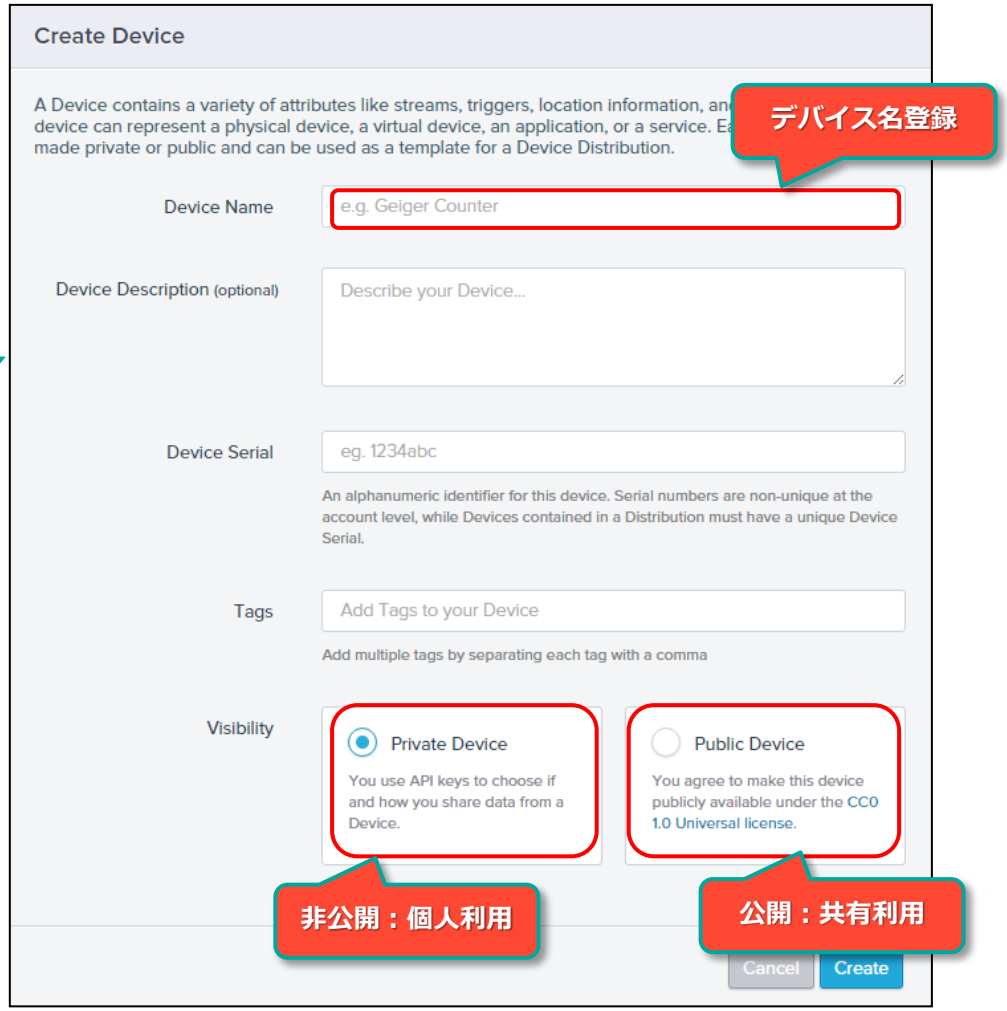


# 3． M2X（AT&T IoTサービス）のID登録

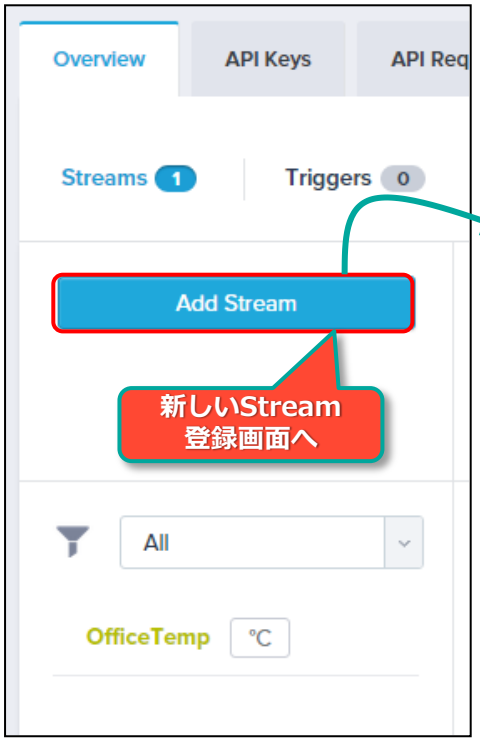


# 4．デバイス（Device）の作成登録

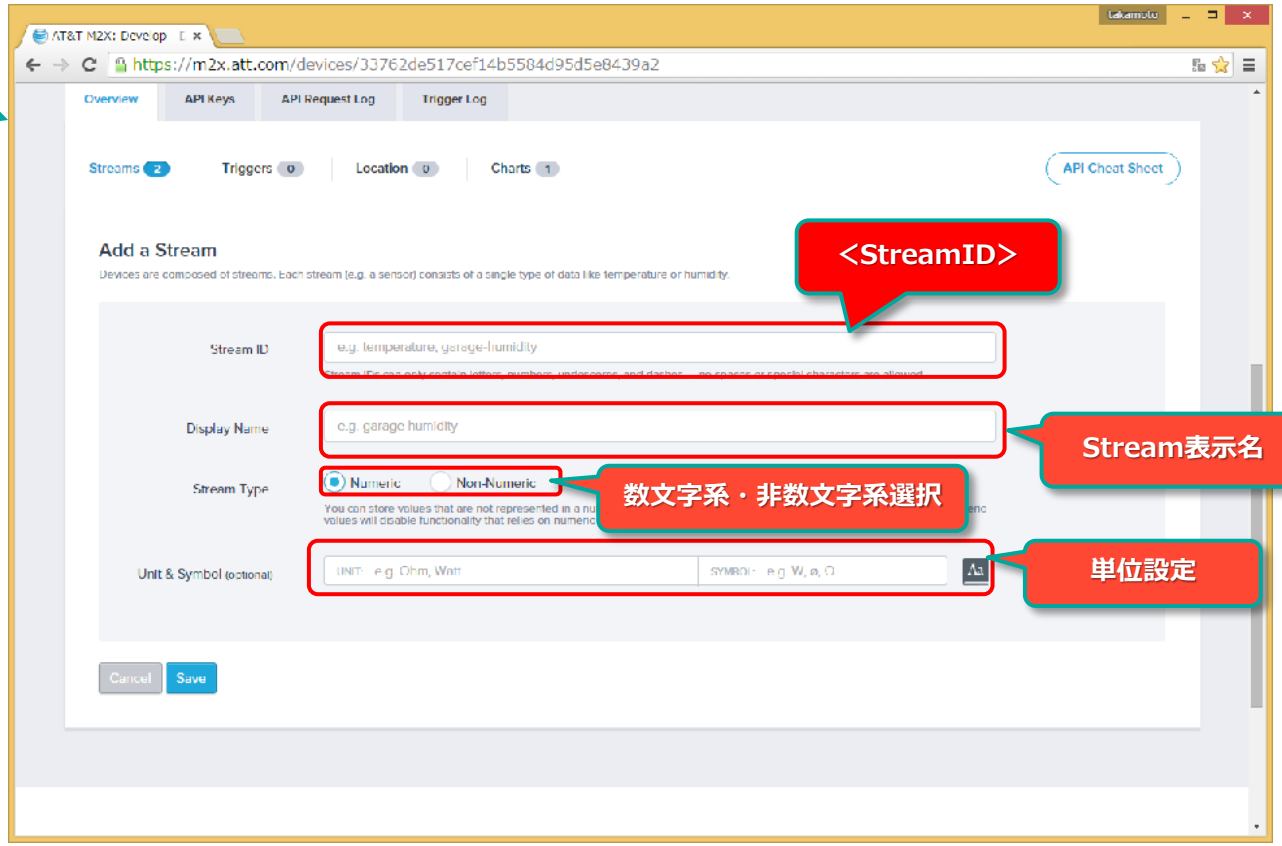

※ deviceIDは、英数文字のキーワードとして
自動的に設定されます。



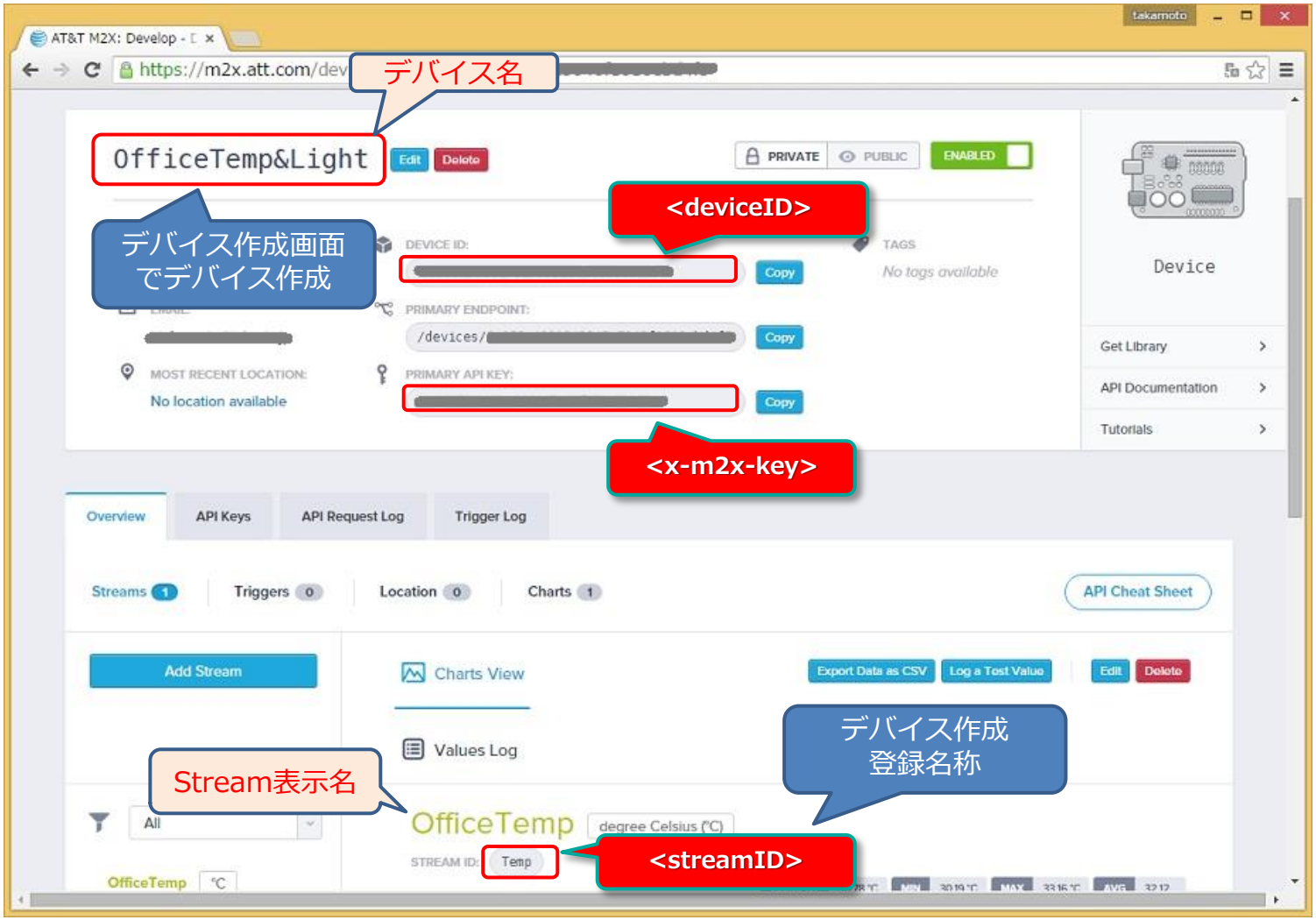
# 5．ストリーム（StreamID）の作成登録




※ StreamIDは、入力した名前が
設定されます



# 6． デバイスIDとスキームIDの登録



# 7．M2Xへのデータアップの書式要件

M2Xへのセンサ値アップは、＄WPコマンドを使って行います。

$WPコマンドを使って、以下の書式の例のような URL と body 、それに header を使って、M2Xクラウドにアップする。

$WP http://api-m2x.att.com/v2/devices/<deviceID>/updates/
"{$"values$" : {$" <streamID>$" : [{ $"timestamp$" : $"<date-time>$" , $"value$" : $" <val>$"}]}} "
"Host: api-m2x.att.com$r$nX-M2X-KEY:<x-m2x-key>$r$nContent-Type:application/json$r$n"

**※ 以下変数の説明**

**<deviceID>** ：デバイスID
**<streamID>** ：ストリームID
**<x-m2x-key>** ：M2Xキー
**<val>** ： データアップするセンサ値
**<date-time>** ： 日時（文字列） 例 “2015-09-20**T**23:55:36**$+**09:00”（$+は特殊文字）

※ 日時は、日本時間を登録（ただしM2Xでの表示は、グリニッジ標準時となる）

# 8．サンプルプログラム①

```
#include <SoftwareSerial.h>
SoftwareSerial iemSerial(4,5);
const unsigned long baudrate = 9600;

#define LIMITTIME 35000 // ms (3G module start time)
```

url,header,body の設定

```
String url    = "http://api-m2x.att.com/v2/devices/<deviceID>/updates/ ";
String header = "¥"Host: api-m2x.att.com$r$nX-M2X-KEY:<x-m2x-key> $r$nContent-Type:application/json$r$n¥"";
String body   = "¥"{¥$¥"values¥$¥" : {¥$¥"<streamID>¥$¥" : [{ ¥$¥"timestamp¥$¥" : ¥$¥"";

//==========================================
void setup() {
  Serial.begin(baudrate);
  Serial.println(">Ready. ¥r¥n Initilaizing...");
  if( _3Gsetup() ) {
     Serial.println("start");
  } else {
     Serial.println(" Connect Error ... Stop");
     while(1);
  }
}
```

setup 3 G初期化

温度センサ値を3分間隔空けてM2Xにアップ

```
void loop () {
  String dtime = datetime();  // Serial.println(dtime); //debug
  float temp = analogRead(A1)*0.488 - 60.0; // TABshield temp  sensor
 if(_3G_WP("$WP " + url + body + dtime + "¥$¥" , ¥$¥"value¥$¥" : ¥$¥"" + String(temp) +"¥$¥"}]}}¥" " +header)){
    Serial.println("Data Update complete:" + iemSerial.readStringUntil('¥n')); }
  else Serial.println("Data Update false...");
  delay(180000); //waiting 3min
}
```

# 8．サンプルプログラム②

```
//============== 3G setup =================
boolean _3Gsetup() {
  pinMode(7,OUTPUT);
  digitalWrite(7,LOW); delay(1000);
  digitalWrite(7,HIGH); delay(100);
  iemSerial.begin(baudrate);
 //------- 3G module begin & connect ---------
  String str;
  unsigned long tim = millis();
  do{  while(!iemSerial.isListening());
       str=iemSerial.readStringUntil('¥n');
  }while(!(str.indexOf("gw3g")>0) && (millis() - tim) <LIMITTIME);
  if( millis() -tim >= LIMITTIME) {
     return false;
  } else return true;
}
//============== $WP command =================
boolean _3G_WP(String command) {
   Serial.println(command); // debu
  iemSerial.println(command);
  String rstr;
  unsigned long tim = millis();  // time set(ms)
  do{ while(!iemSerial.isListening());
       rstr=iemSerial.readStringUntil('¥n');
       Serial.println(rstr);  //debug print....
  }while(!(rstr.indexOf("$WP=")==0) && (millis() - tim) <LIMITTIME);// $WP return check
  return (rstr.indexOf("$WP=")==0);
}
// Get Date & Time (3GIM command)
//  return --> string "2015-12-23T01:23:45%2B09:00"
String datetime() {
  iemSerial.println("$YT");
  while(!iemSerial.available());
  String dtime = iemSerial.readStringUntil('¥n');
  dtime.replace(" ","T"); dtime.replace("/","-");
  return(dtime.substring(7) + "$+09:00");
}
```

3Gシールド用（電源ON）

電源On状態からgw3g文字が返却されるまで待機

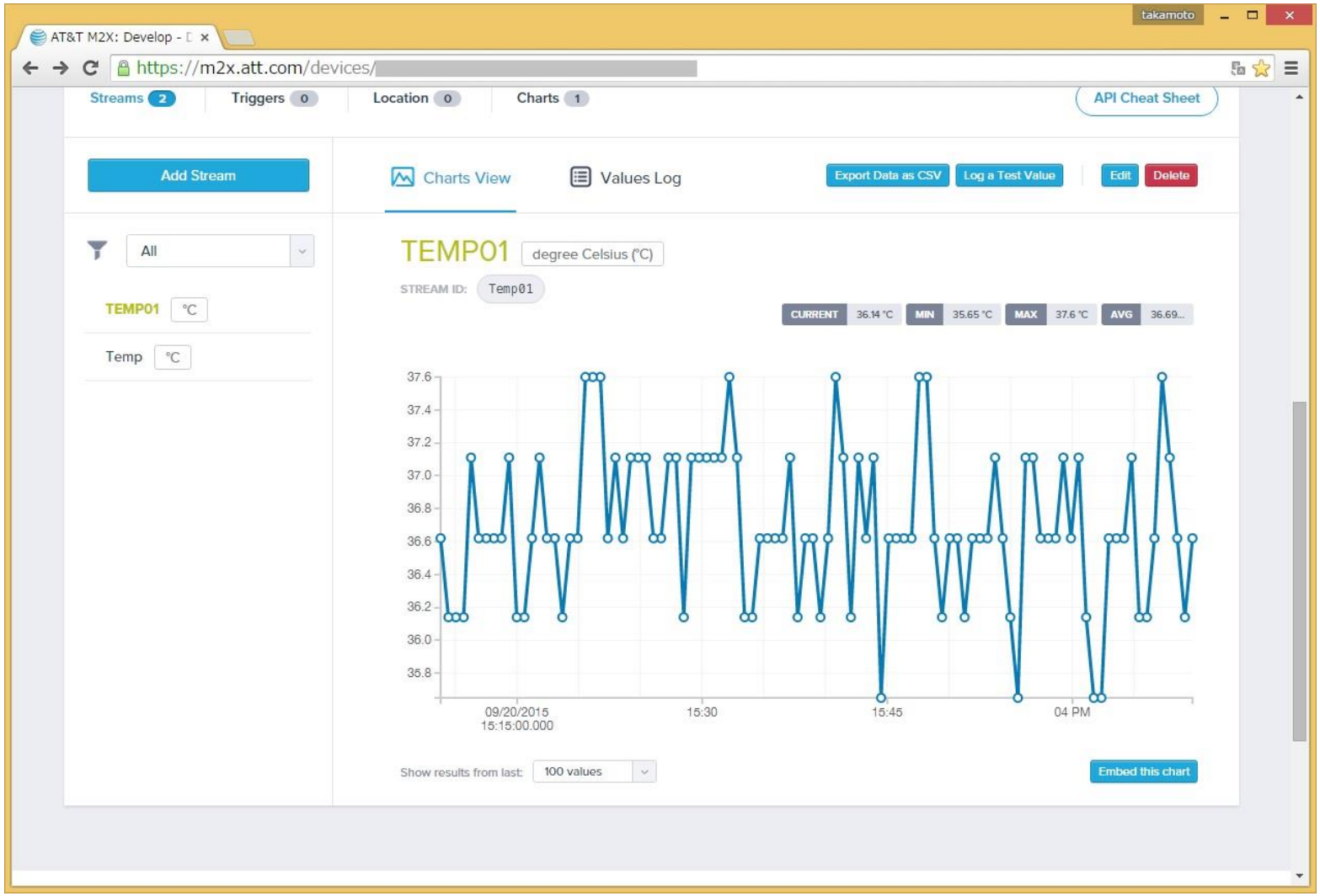
3 G接続状態返却

3 G POST処理

$WPコマンド返却処理

$YTによる時間取得設定機能

# 9．M2Xにデータアップした事例



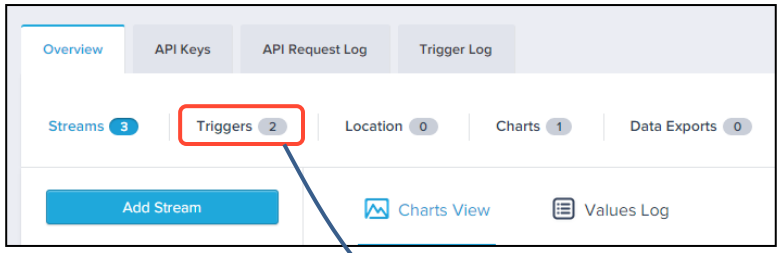
# １０．　M2Xからトリガーでツイートする方法

M2Xにアップしているセンサの値をトリガーにして、ツイートする方法を紹介

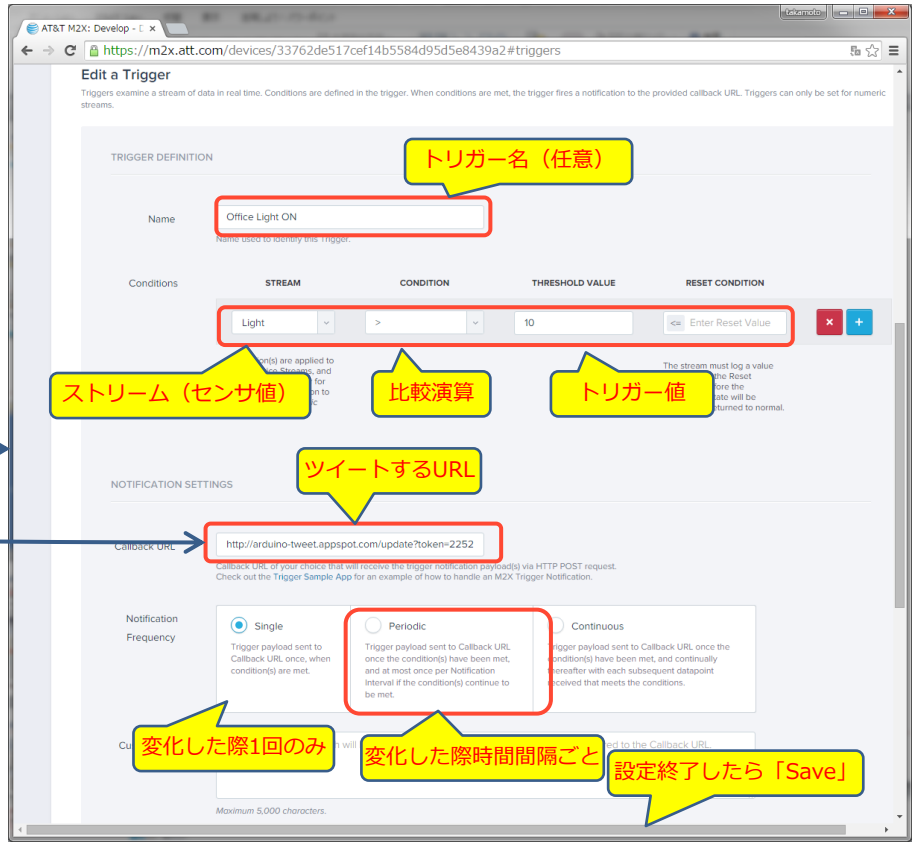
ツイートするURLは、以下の通り

http://arduino-tweet.appspot.com/update?token=**トークン**&status=**ツイート文**

**トリガー設定の選択**




**トリガーの設定**



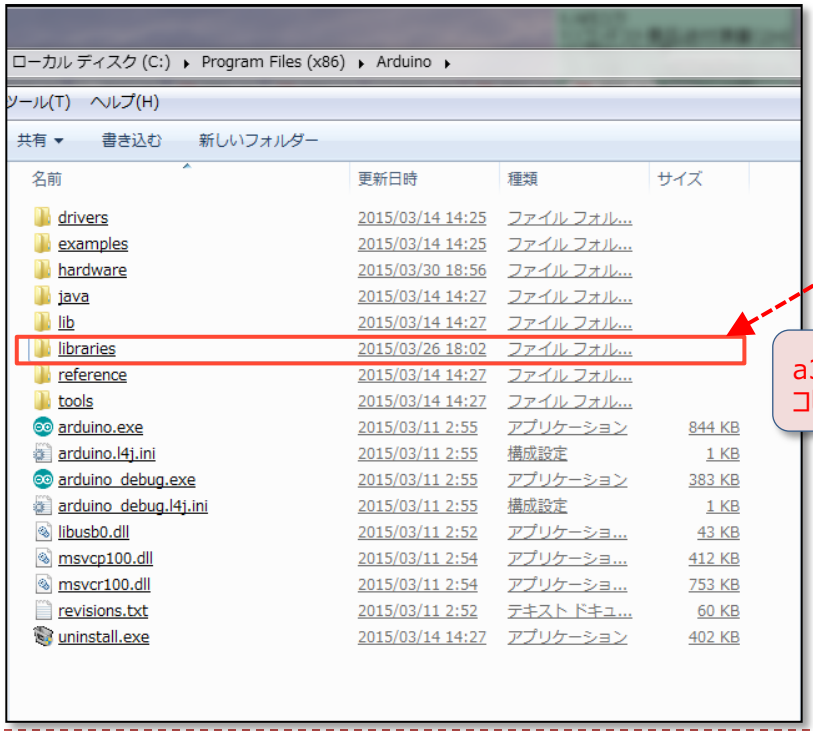

**トリガー**は、M2Xに送られてきたセンサ値の変化を捉え、アクションを起こすものです。
この場合、センサ値がある値より大きいか、小さいかで、URLを起動します。
ここでは、センサ値を見て、ツイッターにツイートするものです。

# 5．Arduino関連ライブラリ（a3gim.zip）

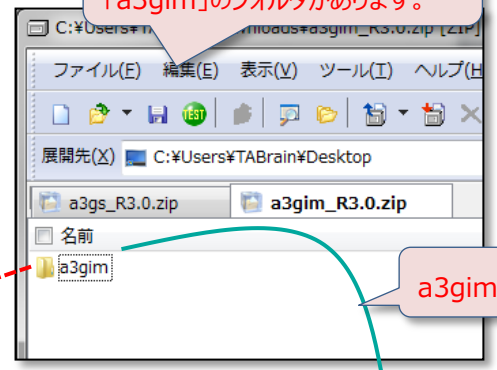
# 1．a3gim_R*.*.zipの提供

**a3gim_R*.*.zipとは**、Arduino上の拡張ボードとして使える**３Gシールドで提供しているa3gs.zipと同じ使い方**ができるArduino上のライブラリ群とサンプルスケッチ群となります。

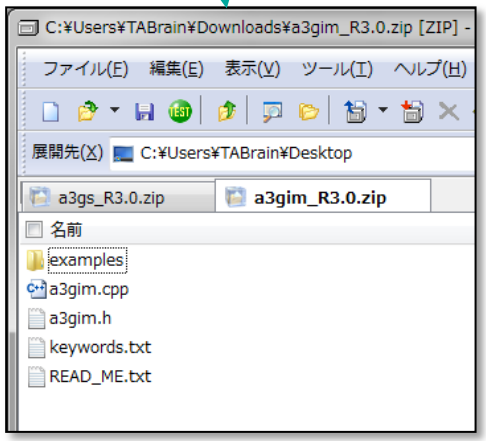
このZIPファイルを、Arduino IDE環境下の「‥¥libraries」配下にコピーしてください。
コピーした後、Arduino IDEを起動すると、メニュー「ファイル」⇒「スケッチの例」に、「a3gim」が表示されます。




a3gim_R*.*.zipを解凍すると「a3gim」のフォルダがあります。




a3gimの内容




a3gimを「libralies」配下にコピーする。

**※これらの使い方は、「３Gシールド」のリファレンスマニュアルを参照ください。**

# 2．a3gimのライブラリを利用する方法

- **概要**
  - 3GIMが提供するa3gimライブラリ機能は、3Gシールドの提供ライブラリとほぼ同等です。
  - そのため、Arduinoと3GIMとの接続を工夫することで、3Gシールド用の下記のライブラリを使用することができます：
    - a3gim　　UNO/Pro用（SoftwreSerialを使用）：３GIM専用に改訂
    - a3gs　　UNO/Pro用（SoftwreSerialを使用）：３Gシールド専用
    - a3gs2　　Mega/Due/Leonardo用（iemSerialまたは3を使用）
- **互いのライブラリの違い**
  - ヘッダファイル(デフォルトのボーレートの違い)
    - a3gim.hのシンボル a3gsBAUDRATE の定義は、「9600」となっています。
    - a3gs.hのシンボル a3gsBAUDRATE の定義は、以前「4800」で、最新では「9600」としています。
    - a3gs2.hのシンボル a3gsBAUDRATE の定義は、「57600」となっています。

【注意】３GIMの出荷時は、9600bpsとしています。

- **3GIMとArduinoとの接続方法**
  - UNOの場合
    - #6をGND、#4を5V、#3をD4、#2をD5、#1を開放(何も接続しない：常時電源ON)、に接続する
  - Mega/Dueの場合（ハードウェアシリアル通信利用）
    - #6をGND、#4を5V、#3をRX3、#2をTX3、#1を開放(何も接続しない：常時電源ON)、に接続する
  - Leonardoの場合（ハードウェアシリアル通信利用）
    - #6をGND、#4を5V、#3をRX1、#2をTX1、#1を開放(何も接続しない：常時電源ON)、に接続する

【補足】ハードウェアシリアル利用のため高速設定可能

【補足】３GIMのボーレートを57600bpsまで高速設定可能

もくじ



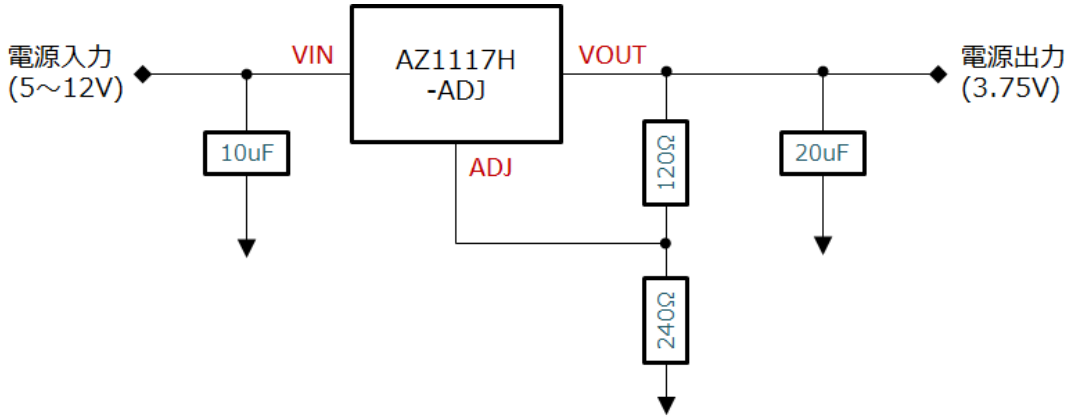
# 補足資料

# 【補足資料1】3GIMコマンド・応答一覧表

| No | 分類 | 機能 | コマンド送信 | 応答（レスポンス）正常受信 | 応答（レスポンス）エラー受信 |
|---|---|---|---|---|---|
| 1 | System | Version | $YV¥n | $YV=OK version¥n | |
| 2 | | RSSI | $YR¥n | $YR=OK rssi¥n | $YR=NG errno¥n |
| 3 | | Serice | $YS¥n | $YS=OK serice¥n | |
| 4 | | IMEI | $YI¥n | $YI=OK imei¥n | $YI=NG errno¥n |
| 5 | | LED | $YL¥n | $YL=OK status¥n | $YL=NG errno¥n |
| 6 | | Baudrate | $YB¥n / $YB baudrate¥n | $YB=OK baudrate¥n | $YB=NG errno¥n |
| 7 | | Reset | $YE¥n / $YE level¥n | $YE=OK level¥n | $YE=NG errno¥n |
| 8 | | Time | $YT¥n | $YT=OK datetime¥n | $YP=NG¥n |
| 9 | | Airplane mode | $YP¥n / $YP mode¥n | $YP=OK mode¥n | $YP=NG errno¥n |
| 10 | | Do Command | $YD password "encrypted-data"¥n | $YD=OK¥n | $YD=NG errno¥n |
| 11 | SMS | Send | $SS msn "message" [encode]¥n | $SS=OK¥n | $SS=NG errno ..¥n / $SS=NG errtype errcode¥n |
| 12 | | Receive | $SR¥n | $SR=OK msn "message"¥n | $SR=NG errno¥n |
| 13 | | Check | $SC¥n | $SC=OK rtn¥n | |
| 14 | GPS | GPS | $LG method¥n | $LG=OK latitude longitude¥n | $LG=NG errno¥n |
| 15 | Web | Get | $WG url ["header"]¥n | $WG=OK nbytes¥nresponse¥n | $WG=NG errno ..¥n |
| 16 | | Post | $WP url "body" ["header"]¥n | $WP=OK nbytes¥nresponse¥n | $WP=NG errno ..¥n |
| 17 | TCP/IP | Read | $TR maxbytes¥n | $TR=OK nbytes¥ndata¥n | $TR=NG errno ..¥n |
| 18 | | Write | $TW "data"¥n | $TW=OK nbytes¥n | $TW=NG errno ..¥n |
| 19 | | Connect | $TC host_or_ip port¥n | $TC=OK¥n | $TC=NG errno ..¥n |
| 20 | | Disconnect | $TD¥n | $TD=OK¥n | $TD=NG errno ..¥n |
| 21 | | Status | $TS¥n / $TS status¥n | $TS=OK status¥n | $TS=NG errno ..¥n |
| 22 | | Get sockname | $TN¥n | $TN=OK ipAddr portNo¥n | $TN=NG errno ..¥n |
| 23 | Profile | Set | $PS profileNum¥n | $PS=OK¥n | $PS=NG errno¥n |
| 24 | | Read | $PR¥n / $PR profileNum¥n | $PR=OK profileNum¥n / $PR=OK apn auth authtype pwd usr dns dns1 dns2¥n | $PR=NG errno¥n |
| 25 | | Reset | $PE profileNum¥n | $PE=OK¥n | $PE=NG errno¥n |
| 26 | Storage | Write | $RW no "data"¥n | $RW=OK¥n | $RW=NG errno ..¥n |
| 27 | | Read | $RR no¥n | $RR=OK nbytes¥ndata¥n | $RR=NG errno ..¥n |

# 【補足資料2】5Vから3.7Vを作り出す回路例

◆ 5～12Vの電源(ACアダプタ等)から3GIMが必要とする3.7V電源を作り出す回路の例を以下に示す：




【図】5Vから3.7Vを出力する電源回路例

◆ 必要な部品は下記の通り：

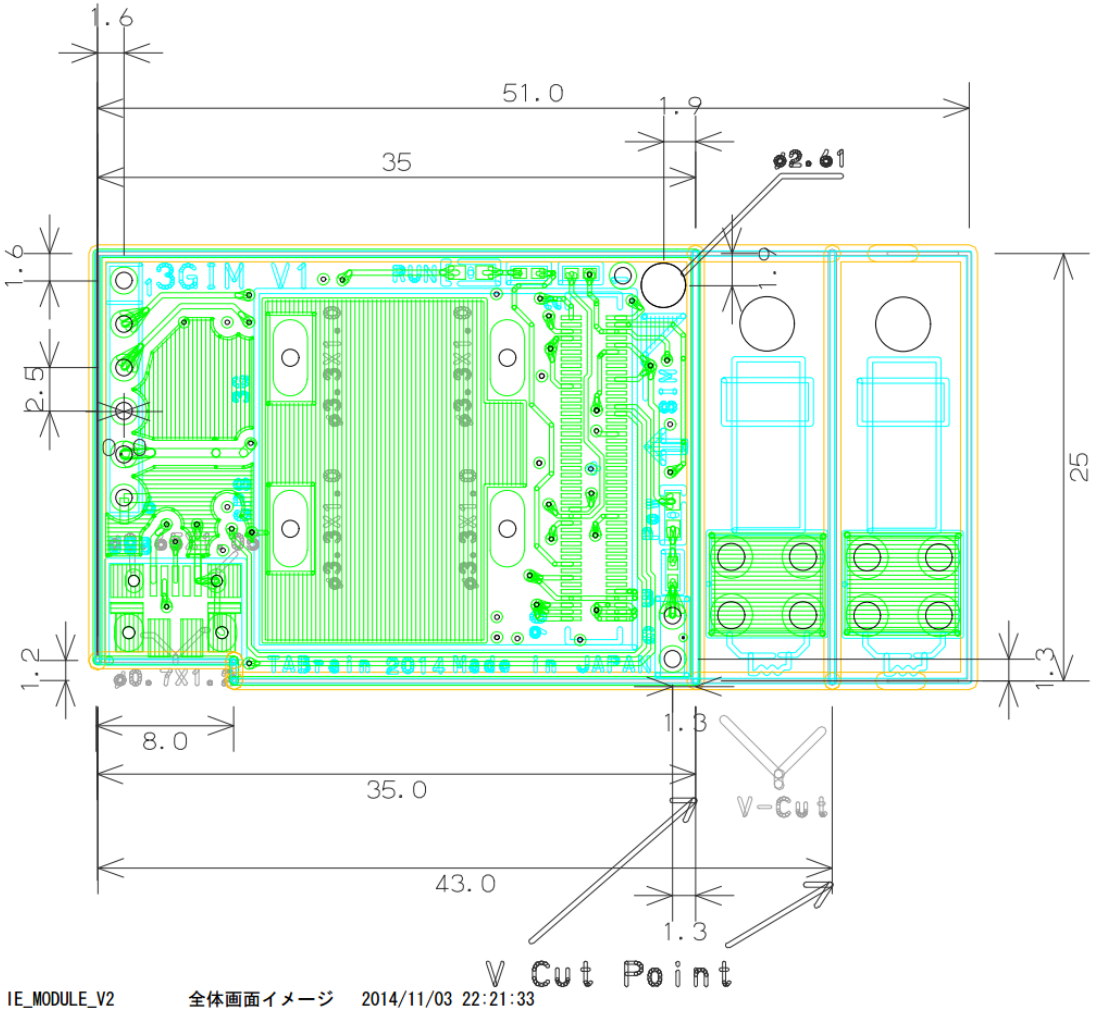
| No | 分類 | パーツ | 数量 | 実売価格(円) | 補足・販売店 |
|---|---|---|---|---|---|
| 1 | 3端子レギュレータ | AZ1117H-ADJ | 1個 | 30 | 秋月電子にて10個単位で販売 |
| 2 | 抵抗 | 1/4W抵抗(240Ω) | 1個 | 10 | 秋月電子・千石電商等で販売 |
| 3 | 抵抗 | 1/4W抵抗(120Ω) | 1個 | 10 | 秋月電子・千石電商等で販売 |
| 4 | 積層セラミックコンデンサ | 25V 10μF | 1個 | 80 | 秋月電子・千石電商等で販売 |
| 5 | タンタルコンデンサ(または積層セラミックコンデンサ) | 10V 22μF | 1個 | 42 | 千石電商等で販売 |

# 【補足資料3】トラブルシューティング

| # | 課題 | 現象 | 対応策 | 補足 |
|---|---|---|---|---|
| 1 | 配線・接続 | ・UART（Tx:送信、Rx:受信）、電源およびGNDが正しく理解できていない | ・３GIMコネクタ部の#1~#6までを正しく理解して上で配線・接続のこと<br>#1（電源On/Off：任意）、#2（RX)、#3（Tx）、#4（3.3V/５V 電源）、#5（3.6~4.2V電源）、#6（GND） | ・#5(VCC)で外部電源を利用する場合には、3.7Vリチウムイオン電池を推奨 |
| 2 | 応答（レスポンス） | ・コマンドを送っても、返信がない<br>・正しい応答でない | ・通信モジュールとマイコンボードとの通信、またはマイコンボードとPCとの通信において以下の原因が考えられる<br>① 3GIMの配線が正しくできていない（配線・接続確認）<br>② 電源供給に問題がある（電源電圧の確認）<br>③ UART通信速度の設定が間違っている（確認設定）<br>④ 初期電源後の待ち時間を考慮不足（３０秒以上待機）<br>⑤ プログラムに間違いがある<br>⑥ Arduino IDE シリアルモニタ画面の改行コード変更 | ・応答が正しく表示されない場合の原因は、配線ミスや配線での接触不良が考えられる<br>・②の電源供給で、VCCの3.6～4.2Vを間違えるケースが多発<br>・⑥の場合、改行選択メニューで「CRおよびLF」を選択のこと |
| 3 | エラー頻発 | ・＃=NGが多発<br>・立ち上げタイミングの問題<br>・電源供給（電流が小さい）問題 | ・配線・接続が正しくできていること<br>・適正なSIMカードの挿入されていること<br>・正しく電源供給できていること | ・RSSI（電波強度測定）やSIMカードのサービス確認<br>・$YRや$YSコマンドで確認 |
| 4 | 時間の取得 | ・時間の取得（＄YT）が間違っている | ・正しいSIMカードとアンテナ接続によって正しく設定される<br>・正しい時間を取得するにはしばらく時間が掛る | ・同 上 |
| 5 | SMS送受信 | ・SMSの送受信ができない<br>・SMSの応答が無い | ・SIMカードが、SMS対応になっていない（切替え必要）<br>・SMSサーバとのやり取りでの不備（何度か読込み必要） | ・同 上 |
| 6 | GPS取得 | ・GPS取得ができない<br>・GPS取得に時間が掛る | ・GPSアンテナが正しく接続されていること<br>・GPS電波状態が良い所（PCから離す）で実施のこと<br>・初期立上げでは数分から10分ほど掛る場合がある<br>・電源供給が正しくできていること | ・一度GPS取得でき、電源が入った状態だと、次からは即取得可能 |
| 7 | ネット接続 | ・Webコマンド群やTCP/IPコマンド群が正しく応答しない | ・３Gアンテナが正しく接続する<br>・正しいSIMカードが挿入されていない<br>・SIMカードの接続不良（再度再挿入などを実施）<br>・電源供給が正しくできていること | 正しいSIMカードとは、Profile設定されたもので、WiKiページで情報公開 |

※その他トラブルが有った場合には、WiKiページにてお問い合わせください。
http://form1.fc2.com/form/?id=816242
基本的なことは、これまでWiKiページサイトや、本資料等にて掲載していますので、そちらをご覧ください。
基本的なことでのお問い合わせは、返答を控えさせていただくことがあります。

# 【補足資料4】外形寸法


1.6
51.0
1.9
35
φ2.61
1.6
2.5
1.2
25
1.3
8.0
35.0
43.0
1.3
V-Cut
V Cut Point
IE_MODULE_V2
全体画面イメージ
2014/11/03 22:21:33

# 【補足資料5】 3GIM サポートサイト

3GIMに関する技術情報が盛りだくさん掲載されています。
http://a3gs.wiki.fc2.com/wiki/3GIMの紹介


**内容（もくじ）**