

本資料は、3 GIM V2.0 + 3 GIMシールド V1.0 を簡単に利用 するためのArduino用ライブラリ**a3gim R4.0** の解説書です。

【注意】本3GIM V2.0モジュールの通信速度は、デフォルトの9600bpsに設定しています(a3gim.hで定義)



3 GIM V2.0+3GIMシールド ライブラリ仕様書 (a3gim R4.0)

2012/10/01~2016/02/25

著作:NPO法人オープンワイヤレスアライアンス

ライブラリ群(a3gim.zip)は、こちらからダウンロードお願い致します。

http://tabrain.jp/3GIM V2.0/a3gim.zip







もくじ

I 編 3 GIM V2.0 + 3 GIMシールドの概要	p.03	p.02
1. 3 GIM V2.0+3GIMシールドとは 2. 3 GIM V2.0+3GIMシールドの構成 II 編 a3gim R4.0ライブラリ概説	p.04	p.05
1. 3GIM V2.0 とは 2. ライブラリ "a3gim R4 0" の概説	p.06 p.07 p.11	·
3. ライブラリ"a3gim R4.0"の使用方法(例) Ⅲ編 a3gim R4.0 ライブラリ 1. コントロール関連の関数	p.13	p.12
 ショートメッセージ関連の関数 Web関連の関数 位置情報取得 (GPS)関連の関数 通信その他機能の関数 TCP/IPの関数 プロファイルの関数 	p.25 p.30 p.36 p.39 p.45 p.57	
8. ATコマンド稼働 IV編 a3gimのインストール	p.60	p.61
1. a3gim.zip ファイルの内容 2. a3gim.zip ファイルの概要 3. a3gimイントール完了の確認	p.62 p.63 p.64	·
V編 サンプル・スケッチ 1. blink led.ino【LEDの点滅】	P.66	p.65
2. check_baudrate.ino【IEM通信モジュールのボーレートの確認サンプル】 3. check_rssi.ino【3 G電波強度取得】 4. check_service.ino【SIMカードのサービス形態取得】 5. get_imei.ino【IEMモジュール端末識別番号取得】 6. get_location.ino【現在位置取得】 7. get_location2.ino【Assisted GPSによる現在位置取得】 8. get_status.ino【通信状態取得 1】 9. get_time.ino【時刻取得 1】 10. get_time2.ino【時刻取得 2】 11. http_get.ino【サーバのレスポンス返却】 12. sample_TCPIP.ino【TCP/IP機能のサンプル】 13. send_sms.ino【ショートメッセージ送信】 14. set_baudrate.ino【通信速度の変更】 15. set_defaultprofile.ino【プロファイルの変更】 16. tweet_sample.ino【ツイートデータ送信】 17. xively_sample.ino【クラウドとの連携】	p.67 p.68 p.69 p.70 p.71 p.72 p.73 p.74 p.75 p.76 P.77 p.79 p.80 p.81 p.82 p.83	
【添付資料.1】a3gim.hライブラリー覧表 【添付資料.2】tweetトークン取得 【添付資料.3】xivelyの利用登録 【添付資料.4】語彙説明 【添付資料.5】電源供給について 【添付資料.6】サンブル1:メール送信 【添付資料.7】サンブル2:xively.com利用		P.87 P.88 P.89 P.92 P.93 P.94 P.95



- 1. 3GIM V2.0+3GIMシールドとは
- 2. 3 GIM V2.0+3 GIMシールドの構成

p.03 p.04

I編. 3GIM V2.0+3GIMシールドの 概要





1. 3GIM V2.0 + 3GIMシールドとは

- ▶ Arduino上で簡単に3GIM V2.0を利用できるようにしたボードが3GIMシールドです。
 - ▶ 3 GIM V2.0は、3 G通信(FOMA)を利用したWeb・TCP通信、SMS送受信、GPS機能等をサポート
 - ▶ NTTdocomo、MVNOのIIJmio、DTI、SORACOM、ほかSO-NET(0SIM)等のSIMカードで利用可能
- メモリサイズの小さいマイコン「Arduino UNO R3」等でも利用できる高機能なライブラリ "a3gim" を提供
- ライブラリが提供する機能は、下記の通り:
 - ▶ Web通信(HTTP/HTTPS GET/POST)
 - TCP/IP通信(connect, disconnect, read, write, status)
 - ▶ 位置情報取得(GPS/Assisted GPS)
 - ショートメッセージ (SMS) 送受信(send, receive, check)
 - ▶ プロファイル確認・設定(APN情報による設定・確認)
 - ▶ **その他**(初期化、電源制御、時刻取得、電波強度取得、IMEI取得、LED制御等)
- ▶ 3GIMシールドを利用するにあたっての留意点は下記の通り:
 - a3gim ライブラリは、Arduino標準ライブラリSoftwareSerialを利用する。
 - ▶ 3GIMシールドでは、シリアル通信UARTは、標準的にはD4(Rx)ピンとD5(Tx)ピンを利用する(変更可能)
 - ▶ D7ピンまたはD8ピンで3GIMの電源On/Offを制御する。

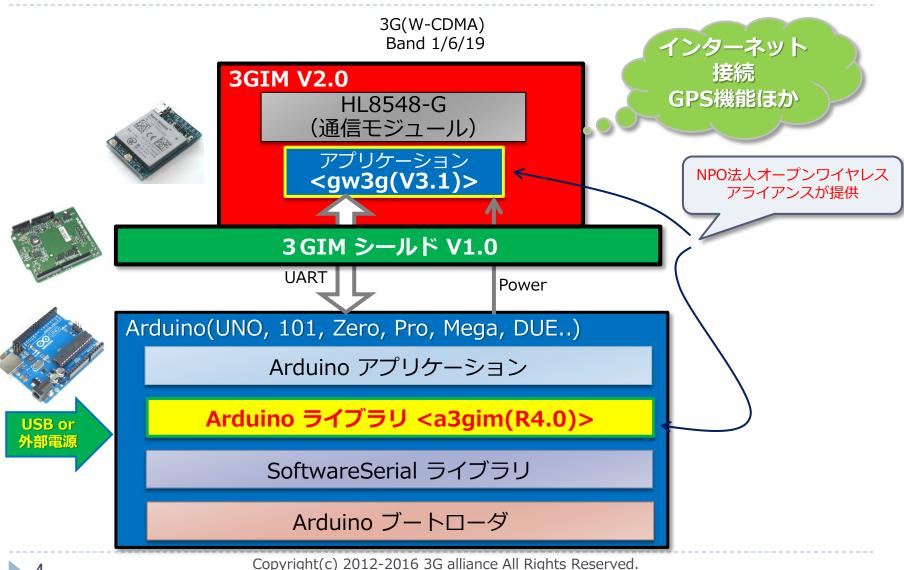








2. 3 GIM V2.0+3 GIMシールドの構成





1. 3 GIM V2.0 とは 2. a3gim R4.0 ライブラリ の概説 p.06 p.07

3. a3gim R4.0 ライブラリ の使用方法(例)

p.11

II編 a3gim R4.0 ライブラリ概説

【注意】関数名の右上に※印があるものは、Arduino GSM/GPRSシールド用ライブラリと互換性がある関数





1. 3GIM V2.0 とは

- ▶ 世界最小サイズの3G通信モジュール・ブレイクアウトボード
 - ▶ シエラワイヤレス社の「HL8548-G」(JATE/TELEC 取得済)・NTTドコモ(IOT取得済)を採用
 - サイズは 35mm × 25mm × 7mm , 重量は7.5 g と非常に小型な通信モジュール
 - ▶ 32ビットARMマイコン(LPC812M101JTB16)を搭載、独自のファームウェアを開発できる
 - ▶ GPS/AGPSを利用可能
 - ▶ さまざまなIoTデバイスやゲートウェイとして利用できる携帯向けで、消費電力が低い

HL8548-Gの主な仕様		
UMTS	Band 1/6/19	
EDGE/GPRS/GSM	850/900/1800/1900 MHz	
GPS	GPS(1575.42MHz) GLONASS(1602MHz)、Assisted GPS	
Speed	7.2Mbps(Download)/5.76Mbps(Upload)	
その他	JATE 取得済み(docomo IOT取得済み)	
サイズ	23mm×22mm×2.5mm	
動作温度	-30℃ ~ 70℃	











2. a3gim R4.0 ライブラリの概説

3Gシールドの制御ライブラリ

分類	メソッド名 ^{※1}	機能概要	補足
	getStatus**	3Gシールドの状態取得	
	begin [*]	ライブラリの初期化	
	end*	ライブラリの終了	
	restart [×]	3Gシールドのリセット	
コントロール	start*	3Gシールドの電源ON	
(Control)	shutdown*	3Gシールドの電源OFF	
(COTTEL OI)	getIMEI	IMEI-IDの取得	
	setLED	LED1のON/OFF	
	setBaudrate	UARTの通信速度の設定	初期は9600bps
	setAirplaneMode	エアプレーンモードのON/OFF	
	getResult	通信結果を取得	※追加
ショート	sendSMS ^{**}	SMSの送信	※仕様変更
メッセージ	availableSMS ^{**}	SMSの受信状態チェック	
(SMS) *	readSMS**	SMSの読出し	
(3.13)	onSMSReceived	SMS着信時のコールバック設定	V4.0では何もしない

- ※ Arduino GSM/GPRSシールド用ライブラリと互換性がある関数です。
- * 利用するSIMカードによって使えない場合があります。









2. a3gim R4.0 ライブラリの概説

3Gシールドの制御ライブラリ

分類	メソッド名	機能概要	補足
	httpGET*	GETメソッドの要求	http/httpsを利用可
Web機能	httpPOST	POSTメソッドの要求	同上
	tweet*	Twitterへの投稿	*
現在位置取得	getLocation	現在位置の取得	緯度経度情報
(GSP)	getLocation2	現在位置の取得 2	緯度経度他情報 ※追加
	getServices	利用可能サービスの取得	
	getRSSI	電波強度の取得	
通信機能その他	getTime	現在時刻の取得	日付・時刻形式
	getTime2	現在時刻の取得	通算秒形式
	getVersion	3Gシールドのバージョンの取得	

- ※ Arduino GSM/GPRSシールド用ライブラリと互換性がある関数
- * 無償サービス「http://arduino-tweet.appspot.com/」を利用(要登録)







2. a3gim R4.0 ライブラリの概説

3Gシールドの制御ライブラリ

分類	メソッド名	機能概要	補足
	connectTCP [*]	TCPコネクションを接続	
	disconnectTCP*	TCPコネクションを切断	
TCP/IP機能	getStatusTCP	TCPコネクション最新状況取得	※追加
107/17/成形	writeBegin	シリアル通信で直接書込み	※追加
	read [*]	データの読込み	2つのバリエーション有
	write [*]	データの書出し	3つのバリエーション有
プロファイル	setDefaultProfile	デフォルトプロファイルを設定	※仕様変更
フロング・ロル	getDefaultProfile	デフォルトプロファイルを取得	※仕様変更
ATコマンド	enterAT	ATコマンドパススルーモード	※追加

※ Arduino GSM/GPRSシールド用ライブラリと互換性がある関数







a3gim R4.0 ライブラリの概説

3Gシールドの制御ライブラリで使用する定数

▶ 各ライブリを利用する上で、留意すべき定数(主に最大値の定義)を下表に示す。これらは、 ヘッダファイル"a3gim.h"で定義されている。

分類	定数名	意味	設定	補足
SMS	a3gimMAX_SMS_LENGTH	SMSメッセージの最大バイト数	120	
SINIS	a3gimMAX_MSN_LENGTH	電話番号の最大バイト数(最大桁数)	11	
	a3gimMAX_URL_LENGTH	URLの最大バイト数	256	%1
	a3gimMAX_HEADER_LENGTH	POSTのヘッダの最大バイト数	512	%1
Web	a3gimMAX_BODY_LENGTH	POSTのボディの最大バイト数	1024	*1
	a3gimMAX_RESULT_LENGTH	GET/POSTのレスポンスの取得可能な最大バイト数	192	%1
	a3gimMAX_TWEET_LENGTH	ツイートメッセージの最大バイト数	60	%1
TCP/IP	a3gimMAX_HOST_LENGTH	ホスト名の最大バイト数	96	%1
TCF/IF	a3gimMAX_DATA_LENGTH	一度に読み書きできるデータの最大バイト数	1024	%2

- ※1 これらの定数は、ATmega328*/32U*(Unoなど)を利用したArduinoではSRAMのサイズが小さい ことからかなり制約が厳しい。ATmega2560/1280(Megaなど)またはADKを利用することで、これ らの最大値を大きくすることができる。
- ※2 大きなデータを読み書きする場合は、複数回に分けてread/writeを実行する。









3. a3gim R4.0ライブラリの使用方法(例)

本ライブラリは、スケッチの中で以下の順序でコントロール用のメソッド(関数)を呼び出すことにより、利用できるようになる:

```
#include "a3gim.h"

void setup()
{

if (a3gSerial.start() == 0) {

// 3Gシールドの電源ONに失敗した
}

if (a3gSerial.begin() == 0) {

// 3Gシールドが使用できるようになった

// 3Gシールドが使用できるようになった

while( a3gSerial.start() == 0 && a3gimの状態を検査

** このようなスケッチでも可能

while( a3gSerial.start() == 0 && a3gSerial.begin() == 0 );
```



1.	コントロール関連の関数	p.13
2.	ショートメッセージ関連の関数	p.25
3.	Web関連の関数	p.30
4.	位置情報取得 (GPS)関連の関数	p.36
5.	通信その他機能の関数	p.38
6.	TCP/IPの関数	p.44
7.	プロファイルの関数	p.53

皿編 a3gim R4.0ライブラリ

本稿では、a3gimライブラリR2.0の使用について説明する。

【注意】関数名の右上に※印があるものは、Arduino GSM/GPRSシールド用ライブラリと互換性がある関数





1. コントロール関連の関数

提供関数ライブラリ

	getStatus*	3Gシールドの状態取得	
	begin [×]	ライブラリの初期化	
	end [*]	ライブラリの終了	
	restart [*]	3Gシールドのリセット	
コントロール	start [*]	3Gシールドの電源ON	
(Control)	shutdown*	3Gシールドの電源OFF	
(COTTEL OI)	getIMEI	IMEI-IDの取得	
	setLED	LED1のON/OFF	
	setBaudrate	UARTの通信速度の設定	初期は9600bps
	setAirplaneMode	エアプレーンモードのON/OFF	
	getResult	通信結果を取得	※追加

Arduino GSM/GPRSシールド用ライブラリと互換性がある関数

概要

a3gimライブラリの初期化・終了、ライブラリの状態の取得、3GIMシールドのリセット、電源ON/OFF、IMEIの取得、 LED1のON/OFF、UARTの通信速度の設定を行う

留意点

- 電源ONには、15秒程度の時間が掛かる。リセットには、5秒程の時間が掛かる。
- 電源OFFには、1秒ほどの時間が掛かる
- setBaudrateによる通信速度の変更には、十分留意すること









1. コントロール関連の関数 getStatus**

int getStat	us(void)
機能概要	現在のライブラリの状態を取得
引数	なし
戻り値	現在の状態(ERROR, IDLE, READY, TCPCONNECTEDCLIENT のいずれか)
補足	各状態の意味は下記の通り: a3gim::ERROR:エラーが発生した a3gim::IDLE:空いている(機能の実行が可能) a3gim::READY:同上 a3gim::TCPCONNECTEDCLIENT:TCPコネクションが接続した

【利用例】

```
int status;
status = a3gSerial.getStatus();
Serial.print("Status is ");
switch (status) {
                                                                            【出力結果例】
 case a3gim::ERROR :
                         Serial.println("ERROR");
                                                    break;
                         Serial.println("IDLE");
                                                                            Status is IDLE
 case a3gim::IDLE:
                                                    break;
 case a3gim::READY :
                         Serial.println ("READY");
                                                    break;
 case a3gim::TCPCONNECTEDCLIENT :
         Serial.println("TCPCONNECTEDCLIENT");
                                                    break;
 default : Serial.println("Unknown"); break;
```







1. コントロール関連の関数 begin ※

● バリエーション1

int begin(c	har* pin)
機能概要	ライブラリを初期化する
引数	pin:未使用(指定は不要)
	0:正常に初期化を実行できた時
戻り値	1: エラーが発生した時(ライブラリは使用不可)
	2 : IEM上のgw3gアプリのバージョンが古い(ライブラリは使用不可)
補足	3Gシールドの電源がONの状態で、本ライブラリの使用に先立って一度だけ本関数を呼び出す必要がある。 初期化に失敗した場合は、end()関数を呼び出して、時間をおいて何度かリトライすることで成功する場合がある。 終了関数end()を呼び出した後は、再度、本関数を呼び出すことができる。 本関数の中では、デフォルトの通信速度で標準ライブラリ「SoftwareSerial」を初期化(begin)する。

【使い方の例】

if (a3gSerial.begin() == 0)
 Serial.println("Succeeded.");









1. コントロール関連の関数 begin ※

● バリエーション2

int begin(c	har* pin, uint32_t baudrate)
機能概要	ライブラリを初期化する
	pin:未使用(指定は不要)
引数	baudrate: 設定する通信速度 (1200/2400/4800/9600/19200/38400/57600/115200)
	0:正常に初期化を実行できた時
戻り値	1:エラーが発生した時(ライブラリは使用不可)
	2: IEM上のgw3gアプリのバージョンが古い(ライブラリは使用不可)
	3Gシールドの電源がONの状態で、本ライブラリの使用に先立って一度だけ本関数を呼び出す必要がある。
補足	初期化に失敗した場合は、end()関数を呼び出して、時間をおいて何度かリトライすることで成功する場合がある。
	終了関数end()を呼び出した後は、再度、本関数を呼び出すことができる。 本関数の中では、指定された通信速度で標準ライブラリ「SoftwareSerial」を初期 化(begin)する。
	通信速度の変更は、setBaudrate()関数を使って事前に行っておく必要がある。

【使い方の例】

if (a3gSerial.begin(0, 9600) == 0)
 Serial.println("Succeeded.");









1. コントロール関連の関数 end[※]

int end(void)		
機能概要	ライブラリの使用を終了する	
引数	なし	
戻り値	0:正常に終了処理を実行できた時	
大ツ心	0以外 : エラーが発生した時	
補足	本関数の中では、標準ライブラリであるSoftwareSerialを終了(end)する。	

【使い方の例】 ・次頁参照







1. コントロール関連の関数 restart[※]

int restart(char* pin)	
機能概要	3Gシールドを再起動(リセット)する
引数	pin:未使用(指定は不要)
戻り値	0:正常にリセットを実行できた時
	0以外 : エラーが発生した時 (リセットできない時)
補足	IEM全体をリセットする。 通常、本関数を呼び出してから10秒程度でIEMはリセット処理を開始し、40秒程度で利用可能な状態となる。 本関数によるリセット後に再度ライブラリを利用する場合は、一旦、終了関数end()を呼び出した後に、初期化関数begin()を呼び出すこと。

【使い方の例】

```
if (a3gSerial.restart() == 0) {
    Serial.println("Restarting..");
    a3gSerial.end();
    if (a3gSerial.begin() == 0)
        Serial.println("I'm OK.");
}
else
    Serial.println("Restart Failed.");
```









1. コントロール関連の関数 start*

int start(char* pin)	
機能概要	3Gシールドの電源をONにする
引数	pin:未使用(指定は不要)
戻り値	0:正常に電源ONを実行できた時
	0以外 :エラーが発生した時(電源ONできない時)
補足	本関数を呼び出しには、40秒程度掛かる(本関数の呼び出しが完了した時点で、3Gシールドが利用可能な状態となっている)。その後、初期化関数begin()を呼び出すことで本ライブラリを利用することができる。

【使い方の例】

```
if (a3gSerial.start() == 0 &&
a3gSerial.begin()) {
 Serial.println("Succeeded.");
    // 成功処理
else
  Serial.println("Restart Failed.");
```









1. コントロール関連の関数 shutdown ※

int shutdown(void)	
機能概要	3Gシールドの電源をOFFにする
引数	なし
戻り値	0:正常に電源OFFを実行できた時
	0以外 : エラーが発生した時(電源OFFできない時)
補足	本関数を呼び出しには、15秒程度掛かる 本関数を呼び出した後は、再度、電源ON関数start()を呼び出すことで3Gシール ドを利用することができる。

【使い方の例】

a3gSerial.end(); a3gSerial.shutdown();









1. コントロール関連の関数 getIMEI

int getIMEI(char* imei)	
機能概要	3Gシールドに装着されているIEMのIMEIを取得する
引数	imei : 取得したIMEI(サイズは a3gimIMEI_SIZE バイト)
戻り値	0:正常に取得できた時
	0以外: エラーが発生した時 (取得できない時)
補足	IMEIとは3G通信モジュール(IEM)の識別IDである(電話番号とは無関係) 引数imeiが指す結果格納場所のスペース(a3gimIMEI_SIZE バイト=16桁)は、 あらかじめ呼び出し側で確保しておくこと。

【使い方の例】

```
char imei[a3gimIMEI_SIZE];
if (a3gSerial.begin() == 0) {
   a3gSerial.getIEI(imei);
   Serial.println(imei);
}
```



【出力結果例】

354563020267950

このIMEIの中に、IEMモ ジュールに記載されたIDが 含まれています。









1. コントロール関連の関数 setLED

int setLED(boolean sw)	
機能概要	3Gシールドに搭載されているLED1を制御する
引数	sw: ONにする時はTRUE、OFFにする時はFALSEを指定する
戻り値	0:正常に設定できた時
	0以外 : エラーが発生した時
補足	LED1(緑色のLED)が3Gシールドのどこの位置に配置されているか等は、「取扱説明書」を参照のこと。

【使い方の例】

```
if (aFlag) {
   a3gSerial.setLED(TRUE);
   led1_status = TRUE;
   Serial.println("LED1 is turned on.");
}
```







1. コントロール関連の関数 setBaudrate

int setBaudrate(int baudrate)		
機能概要	Arduinoと3Gシールドを仲介するUARTの通信速度を設定する	
引数	baudrate :設定する通信速度(2400/4800/9600/19200/38400/57600の いずれか)	
戻り値	0:正常に変更できた時	
大り1世	0以外 : エラーが発生した時	
補足	本関数の利用には十分留意すること。不適切な値を設定した場合は、3Gシールドを利用することができなくなる可能性がある。 工場出荷時の通信速度は、安定動作が可能な 9600(bps) となっている。 通信速度をデフォルトの設定値よりも高くするには、ハードウェアシリアルの利用を推奨する。 本関数による通信速度の変更は、直ちに有効となる。 デフォルトの通信速度と異なる通信速度を設定した場合は、次回の初期化の際には通信速度を指定してbegin()を呼び出すこと(詳細はbegin()の項を参照)	

【使い方の例】

```
if (a3gSerial.setBaudrate(9600) == 0) {
    Serial.println("Baudrate was changed.");
    Serial.println("Please reset me now.");
}
```

注意:設定した通信速度をスケッチ内で呼び出す必要があります。 a3gim.h のスケッチ内の「a3gimBAUDRATE」の設定となります。 間違った場合には、サンプルスケッチの「check_baudrate.ino」 で確認してみてください。









1. コントロール関連の関数 setAirplaneMode

int setAirplaneMode(boolean sw)	
機能概要	3Gシールドのエアプレーン(機内)モードを制御する
引数	sw :ONにする時はTRUE(=1)、OFFにする時はFALSE(=0)を指定する
戻り値	0:正常に設定できた時
	0以外 : エラーが発生した時
補足	エアプレーンモードがONの時は、SMSを含むすべての3G通信は実行できない。 エアプレーンモードをONにすることで、消費電力を大幅に節約することができる。 リセットまたは電源のOFF/ONで、デフォルトの設定(OFF)に戻る。

【使い方の例】

```
if (aFlag) {
   a3gSerial.setAirplaneMode(true);
   airplaneMode = true;
   Serial.println("Airplane mode on");
}
```







2. ショートメッセージ関連の関数

提供関数ライブラリ

分類	メソッド名 ^{※1}	機能概要	補足
> _ 1	sendSMS**	SMSの送信	
ショート メッセージ	availableSMS**	SMSの受信状態チェック	
(SMS)	readSMS**	SMSの読出し	
(31.13)	onSMSReceived	SMS着信時に呼び出す関数を設定	V4.0では何もしない

▶ 概要

- ▶ 3GのSMS (Short Message Service: 120文字程度までの簡易メッセージングサービス) を利用する
- ▶ SMSは、通信キャリアにまたがって送受信できる

留意すべき点

- ▶ 使用するSIMカード(通信サービス)により、SMSが利用できない場合がある
- ▶ 通信料金(送信側に課金、一般には定額プランの範囲外)に注意すること
- ▶ 通信回線の状態によっては、SMSの配送遅延が起こる場合がある(常に即時配送できるとは限らない)
- ▶ 3Gネットワークとの相性により、何度も同一のSMSを受信してしまう現象が発生する場合がある
- ▶ SMSが届かない場合は、受信側のSMS受信拒否設定にも注意すること
- ▶ SMSのメッセージの中に利用できない文字が存在することに注意すること(例えば、"@"文字等は使えない)







2. ショートメッセージ関連の関数 sendSMS[※]

int sendSMS (const char* to, const char* msg, int encode)	
機能概要	SMS(ショートメッセージ)を指定した宛先へ送信する
	to: 送付先の電話番号(ハイフォンなしの10ケタの半角数字)
引数	msg: 送信するメッセージ(最大a3gimMAX_SMS_LENGTH文字)
JISA	encode :メッセージに使用するエンコード方法(a3gimCS_* で指定)。省略可 (省略時はASCIIエンコードと解釈)
戻り値	0:正常に送信できた時
	0以外 : 送信できなかった時
補足	本機能を利用するためには、SMSが利用できる通信サービス(SIMカード)を利用する必要がある。 関数getServices()にて、 a3gimSRV_CS または a3gimSRV_BOTH のいずれかの返却値が取得できる場合にはSMSは利用できる。 3 GIM(V2.0)では、日本語のSMSは取り扱いできない。







2. ショートメッセージ関連の関数 available SMS ※

boolean availableSMS (void)	
機能概要	SMSが届いているかをチェックする
引数	なし
戻り値	true: SMSが届いている時
	false: SMSが届いていない時
補足	

【使い方の例】

```
char msg[a3gimMAX_SMS_LENGTH+1], msn[a3gimMAX_MSN_SIZE+1];
void loop() {
    if (a3gSerial.availableSMS()) {
        int msgLen = sizeof(msg);
        int msnLen = sizeof(msn);
        a3gSerial.readSMS(msg, msgLen, msn, msnLen);
        // 受信したSMSの処理
    }
    delay(1000);
}
```





2. ショートメッセージ関連の関数 readSMS[※]

boolean readSMS (char* msg, int msglength, char* number, int nlength)	
機能概要	受信したSMSを読み出す
	msg: [OUT] 読み出したメッセージ(最大a3gimMAX_SMS_LENGTH文字)
	msglength: msgのサイズ(バイト数)
引数	number: [OUT] SMSの送信元の電話番号
	nlength: numberのサイズ(バイト数) (通常、a3gimMAX_MSN_LENGTH文字を確保)
戻り値	true:正常に読み出した時
	false: 読み出し不可の時(SMSを未受信、SMS使用不可 等)
補足	正常に読み出せた場合は、msgおよびnumberは'¥0'文字で終端される。 メッセージはASCIIで読み出すことができる。 上記以外は、sendSMS()を参照







2. ショートメッセージ関連の関数 onSMSReceived

int onSMSReceived (void (*handler)(void))		
機能概要	SMSが着信した時に呼び出される関数を設定する(V4.0では何もしない)	
引数	handler:呼び出される関数(戻り値・引数は無)へのポインタ	
戻り値	0:正常に設定できた時	
	0以外 :設定できなかった時	
補足	本関数は旧バージョンと互換性を維持するために残しているものである。	







3. Web関連の関数

提供関数ライブラリ

分類	メソッド名	機能概要	補足
Web機能	httpGET*	GETメソッドの要求	http/httpsを利用可
	httpPOST	POSTメソッドの要求	
	tweet*	Twitterへの投稿	*

- ※ Arduino GSM/GPRSシールド用ライブラリと互換性がある関数
- * 無償サービス「http://arduino-tweet.appspot.com/」を利用(要Twitterの登録)

▶ 概要

- ▶ http/httpsを簡単に利用できる。
- ▶ GET/POSTメソッドを利用できる。
- ▶ Web機能の関数は、すべて同期処理である。そのため、レスポンスが取得できるまで、あるいは通信がタイムアウト(30秒程度)するまで呼び出し元には制御は戻らない。
- ▶ tweetは、サードパーティのフリーサービスを利用することで使用できる(ユーザ登録が必要、利用条件はそのサービスに従う)。
 - ▶ 詳細は http://arduino-tweet.appspot.com/ (3Gシールドアライアンスとは関係のないサービス)

留意点

- ▶ 使用するSIMカードで、3Gパケット通信が利用できること
- ▶ 通信料金(http/https通信の利用は、通常、定額プランの範囲内)に留意すること
- ▶ 日本語の取り扱いにはご注意ください。リクエストを送る相手サーバにより、日本語の文字コードが決まりますが、Arduinoでは日本語の処理を簡単に記述することができません。英語のみを取り扱うことを推奨します。





3. Web関連の関数 httpGET ※

int httpGET (const char* server, uint16_t port, const char* path, char* result, int resultlength, boolean ssled=false, const char* header=NULL)

resultlength, boolean ssled=false, const char* header=NULL)			
機能概要	指定したサーバ、ポート、パスに対して、httpまたはhttps/GETリクエストを発行して、 そのレスポンスを返却する		
引数	server: サーバのドメイン名またはIPアドレス		
	port : サーバのポート番号 (通常は80を指定)		
	path: URLのパス		
	result: [OUT] レスポンスの格納先(スペースは呼び出し側で確保)		
	resultlength: resultのサイズを指定		
	ssled: httpsを利用する場合はtrue、httpを利用する場合はfalseを指定(省略可能で、省略時はfalseと同じ)		
	header :特殊なヘッダの指定(最大a3gimMAX_HEADER_LENGTHバイト)		
戻り値	0 : 正常にGETできた時		
	0以外 : GETできなかった時		
補足	本関数の実行時間は、状況によって最大30秒程度掛かる。 サーバからのレスポンスのサイズが大きい場合は、resultlength以降のデータは破棄される。 resultは、'¥0'文字で終端される。文字コードは、接続先のサーバに依存する。 また、レスポンスにはヘッダは含まれない(ボディ部分のみ)		







3. Web関連の関数 httpGET **

補足

特にヘッダの指定がない場合は、リクエストのヘッダは下記の通りである:

Host: server:port

ヘッダを指定する場合は、下記のように指定する。2つ以上のヘッダを指定する場合は、それらの間を「\$r\$n」で区切り、末尾も「\$r\$n」で終わる(終端の改行は省略可能)。
 char *header = "Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==\$r\$nX-Myheder: XYZ\$r\$n"

path引数にはクエリー文字列を含めることができるが、事前にURLエンコードを施しておく 必要がある。また、httpGET()関数では'\$'文字を特殊扱いするため、'\$'文字を文字列に含め る時は、httpPOST()で解説しているようなエスケープシーケンスに従うこと。









3. Web関連の関数 httpPOST

int httpPOST (const char* server, uint16_t port, const char* path, const char* header, const char* body, char* result, int* resultlength, boolean ssled=false)			
機能概要	指定したサーバ、ポート、パスに対して、httpまたはhttps/POSTリクエストを発 行して、そのレスポンスを返却する		
引数	server: サーバのドメイン名またはIPアドレス		
	port:サーバのポート番号(通常は80を指定)		
	path: URLのパス		
	header: HTTPのヘッダ文字列(最大a3gimMAX_HEADER_LENGTHバイト)		
	body : HTTPのボディ文字列(最大 a3gimMAX_BODY_LENGTH バイト)		
	result:[OUT]レスポンスの格納先(スペースは呼び出し側で確保)		
	resultlength: [IN/OUT] resultのサイズを指定、呼び出し結果のサイズが返却		
	ssled: httpsを利用する場合はtrue、httpを利用する場合はfalseを指定(省略可能で、省略時はfalseと同じ)		
戻り値	0 :正常にPOSTできた時		
	0以外 : POSTできなかった時		









3. Web関連の関数 httpPOST



int httpPOST (const char* server, uint16_t port, const char* path, const char* header, const char* body, char* result, int* resultlength)

引数headerでは、HostとContent-Lengthの指定は不要である。

引数bodyでは、最後の空行は指定不要である。

本関数の実行時間は、状況によって最大30秒程度掛かる。

サーバからのレスポンスのサイズが大きい場合は、resultlength以降のデータは破棄される。

resultは'¥0'文字で終端される。文字コードは、接続先のサーバに依存する。

レスポンスには、ヘッダは含まれない(ボディ部分のみ)

引数headerおよびbodyでは、下記の'\$'文字を使ったエスケープシーケンスをサ

ポートする。直接、制御文字を指定することはできないので注意すること:

補足

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(スケッチでの文字列における"0xhh"と同義) 引数headerやbodyでは、バイナリデータをそのまま取り扱うことはできない。また、レスポンスは文字列として返却するため、'¥0'文字を含むことはできない。バイナリデータを透過的に扱った通信を行いたい場合は、TCPIP関数を利用する。







3. Web関連の関数 tweet **

int tweet (const char* token, const char* msg)		
機能概要	Twitterへ投稿する	
引数	token: アクセスに必要なトークン(認証情報)	
	msg:投稿するメッセージ(最大a3gimMAX_TWEET_LENGTHバイト)	
戻り値	0:正常に投稿できた時	
	0以外 :投稿できなかった時	
補足	tweetは、下記のフリーサービスを利用することで使用できる。ユーザ登録が必要で、利用条件はこのサービスに従う。詳細は http://arduino-tweet.appspot.com/を参照のこと。 【注意】上記サービスの制限により、同一メッセージを連続して投稿することはできない。また、一定時間内に投稿できるメッセージ数に制限がある(2012/10時点の制限では、1分間に1回の投稿まで)。この制限を守らない場合は、正しく引数を指定している場合でも本関数はエラーを返却する。	







4. 現在位置取得(GPS)関連の関数

▶ 提供関数ライブラリ

分類	メソッド名	機能概要	補足
用大位等取得(CDC)機能	getLocation	現在位置の取得	内蔵GPSを使用
現在位置取得(GPS)機能	getLocation2	現在位置の取得2	

▶ 概要

- ▶ HL8548-G内蔵GPSや3Gネットワークを利用して位置を測位
- 引数の指定により、下記のいずれかの測位方法を選択できる:
 - MPBASED
 - □ GPSを利用して現在位置を測位する。GPSが利用できない場合は、3Gネットワークを利用する。
 - MPASSISTED
 - □ 3Gネットワーク上のロケーションサーバを利用して現在位置を測位する。
 - MPSTANDALONE
 - □ GPSのみを利用して現在位置を測位する。

留意点

- ▶ 通信サービス(例えば、IIJmio等)によっては、3Gネットワーク上のロケーションサーバを利用することができない。その場合は、GPS単独の測位のみが利用できる。
- 測位方法としてa3gimMPSTANDALONEを指定した場合は、通信料金(通常、定額プランの範囲内だが、 SIMカードの通信サービスによる)が発生する
- ▶ 屋内や都心等のように、上空にある衛星の電波がGPSアンテナで補足できない場所では、正しく測位できない場合がある。







4. 現在位置取得 (GPS) 関連の関数 getLocation

int getLoca	tion(int method, char* latitude, char* longitude)
機能概要	現在位置を取得する
	method:測位方法(a3gimMPBASED/a3gimMPASSISTED/a3gimMPSTANDALONEのいずれか)を指定
引数	latitude: [OUT] 緯度(北緯) dd.dddddd形式、ただし桁数は場合により可変
	longitude: [OUT] 経度(東経) ddd.dddddd形式、同上
戻り値	0:正常に取得できた時
人り他	0以外 :取得できなかった時(latitude, longitudeの値は不定)
補足	本関数の実行には、数十秒~3分程度の時間が掛かる。 AGPSサーバは、3GIM(V2.0)ではGoogleのサーバを利用する(今後変更となる可能性がある) 本関数は、同期処理である。そのため、測位処理が完了するまで、あるいは測位が 失敗するまで呼び出し元には制御は戻らない。





4. 現在位置取得 (GPS) 関連の関数 getLocation2

int getLocation2(char* latitude, char* longitude, char *height, char *utc, int *quality, int *number) 現在位置を取得する 機能概要 latitude:[OUT] 緯度(北緯:度) dd.dddddd形式、ただし桁数は場合により可変 longitude: [OUT] 経度(東経:度) ddd.dddddd 同上 **height**: [OUT] アンテナの海抜高さ(単位:m) 引数 utc: [OUT] 協定世界時での時刻(hhmmss) **quality**: [OUT] 位置特定品質。0 = 位置特定できない、1 = SPS(標準測位 サービス)モード、2 = differenctial GPS(干渉測位方式)モード number: [OUT] 使用GPS衛星数 **0**:正常に取得できた時 戻り値 **0以外**:取得できなかった時(latitude, longitudeの値は不定) 本関数の実行には、数十秒~3分程度の時間が掛かる。 AGPSサーバは、3 GIM(V2.0)ではGoogleのサーバを利用する(今後変更となる 補足 可能性がある) 本関数は、同期処理である。そのため、測位処理が完了するまで、あるいは測位が



失敗するまで呼び出し元には制御は戻らない。





5. 通信その他機能の関数

提供関数ライブラリ

分類	メソッド名	機能概要	補足
	getServices	利用可能サービスの取得	
	getRSSI	電波強度の取得	
通信その他機能	getTime	現在時刻の取得	日付・時刻形式
	getTime2	現在時刻の取得	通算秒形式
	getVersion	3Gシールドのバージョンの取得	

留意点

▶ 時刻は、使用している3Gネットワークを介してインターネット上のサーバから取得する(タイムゾーンや時刻精度は利用するネットワークに依存する)







5. 通信その他機能の関数 getServices

int getServices(int& status)		
機能概要	現在利用できるネットワークサービスを取得する	
引数	status: [OUT] 利用できるネットワークサービス(下記のいずれか) a3gimSRV_NO(=0): サービス利用不可 a3gimSRV_PS(=1): パケット通信サービスのみ a3gimSRV_CS(=2): 音声通信(+SMS)サービスのみ a3gimSRV_BOTH(=3): パケット通信、音声通信(SMS)いずれも可	
戻り値	0:正常に取得できた時	
次ツ旭	0以外 : 取得できなかった時(statusの値は不定)	
補足	3GIM(V2.0)では、 a3gimSRV_CS と a3gimSRV_BOTH は返却しない。つまり、SMSが利用できるかどうかは判断できない。	

【使い方事例】

```
int status;
if (a3gim.getServices(status) == 0)
Serial.println(status);
```









5. 通信その他機能の関数 getRSSI

int getRSSI(int& rssi)		
機能概要	3Gの電波強度(RSSI)を取得する	
引数	rssi: [OUT] 取得した電波強度(単位はdBm) <範囲:-1 ~ -113>	
戻り値	0:正常に取得できた時	
大ツ他	0以外 : 取得できなかった時 (rssiの値は不定)	
補足	電波強度は必ずマイナス値が返却される。0に近いほど電波強度は強い。	

【電波受信レベルの測定サンプル】

実際に測定した実績では、3Gアンテナを付けないで測定した場合では「-113dBm」を計測、 また電波状態の良いところでは「-68dBm」程度を計測できている。

【使い方事例】

```
int rssi;
if (a3gim.getRSSI(rssi) == 0)
    Serial.println(rssi);
```









5. 通信その他機能の関数 getTIME

int getTime(char* date, char* time)		
機能概要	現在の日付・時刻を取得する	
引数	date: [OUT] 取得した日付("YYYY-MM-DD"形式)	
り女人	time: [OUT] 取得した時刻("HH:MM:SS"形式)	
= 6 は	0:正常に取得できた時	
戻り値	0以外 : 取得できなかった時 (date/timeの値は不定)	
補足	日付・時刻は使用している3Gネットワークを介して日本のサーバから取得するため、精度およびタイムゾーンはネットワークに依存する。(日本国内で利用する場合は、タイムゾーンは日本(JST)となる) 時刻は24h制(固定)である。ただし、自動調整出力となるため、変更は不可。	

【使い方事例】

```
【利用例】
if (a3gim.getTime(date, time) == 0) {
   Serial.print(date);
   Serial.print(" ");
   Serial.println(time);
```



【出力結果例】

2016/02/25 12:10:43

【注意:時刻自動調整機能について】

長く電源を入れていないと、時刻が一旦1980年1月5日16:00:00に戻る。 電源を入れて動かしている間に、一時的に現在時刻より16時間遅れで表示され、 さらに、日本時間で表示されるようになる。







5. 通信その他機能の関数 getTIME2

int getTime2(uint32_t& seconds)		
機能概要	現在の時刻(1970/1/1からの通算秒:「UNIX時間」とも言う)を取得する	
引数	seconds: [OUT] 取得した通算秒	
戻り値	0:正常に取得できた時	
	0以外 : 取得できなかった時 (date/timeの値は不定)	
補足 時刻を秒単位で取得できるため、時刻の比較処理等が簡単となる。 日付・時刻の精度およびタイムゾーンについては、getTime()を参照。 秒への換算処理では、閏(うるう)年も考慮している。 2038年*問題が発生する可能性がある。		

※ **2038年問題**は、ISOの通算秒の定義に1970年1月1日からとしていて、C言語の標準で32ビット符号付intを採用している場合、2038年1月19日3時14分7秒(UTC、以下同様)を過ぎると、この値がオーバーフローし、負と扱われるため、コンピュータが誤動作する可能性があるとされる問題。【詳細はウィキペディア参照】









5. 通信その他機能の関数 getVersion

int getVersion(char* version)		
機能概要	3 GIM ファームウェアgw3gアプリのバージョンを取得する	
引数	version : [OUT] 取得したバージョン("9.9"形式)	
戻り値	0:正常に取得できた時	
	0以外 : 取得できなかった時(versionの値は不定)	
補足	begin()の処理の中で3Gシールドのバージョンと本ライブラリの整合性をチェックしているため、通常、本関数を利用する必要はない。	







6. TCP/IPの関数

・ 提供関数ライブラリ

分類	メソッド名	機能概要	補足
	connectTCP*	TCPコネクションを接続	
	disconnectTCP*	TCPコネクションを切断	
TCP/IP	getStatusTCP	TCPコネクション最新状況取得	
TCP/IP	writeBegin	シリアル通信で直接書込み	
	read*	データの読込み	3バリエーション有
	write*	データの書出し	3バリエーション有

【注意事項】

※ Arduino GSM/GPRSシールド用ライブラリと互換性がある関数

- TCP/IP v4のみサポートする。
- 一度に一つのコネクションだけを利用できる。(Web機能とは独立して使用できる)
- 本機能で提供する関数では、すべて同期的に処理する。そのため、サーバから結果が得られるまで、 エラーが発生するまで、あるいは通信がタイムアウトするまで呼び出し元には制御が戻らない。
- 接続や通信では、タイムアウト時間として30秒が設定されている
- readやwriteでエラーが発生した時は、接続(connectTCP)からやり直す必要がある
- TCP/IP機能を利用するためには、利用するSIMカードでパケット通信が利用できる必要がある。 また、契約プランによっては通信料金が高額になる場合があるため注意すること
- 利用するSIMカードによっては、使用できるポート番号に制限(80番のみ等)がある場合がある。
- 1バイト単位でのread/writeは、実行効率が悪く、かつ実行速度が遅い。そのため、 できるだけ複数バイト単位でread/write関数を呼び出して処理することが望ましい。
- 一部のread/write関数で、バイナリデータを透過的に取り扱うことができる。(Ver2.0以降)
- read関数がノンブロッキング(読み出すべきデータがない場合は、待たずに直ちに呼び出し元へリターンする)
- で動作するように変更した。(Ver2.0以降)









6. TCP/IP機能の関数 connectTCP

int connectTCP(const char* server, int port)		
機能概要	指定したサーバ、ポート番号へ接続して、TCPコネクションを確立する	
引数	server:接続するサーバのホスト名またはIPアドレス port:接続するポート番号	
戻り値	0:正常に接続できた時	
	0以外 : エラーが発生した時(戻り値はエラー番号を表す)	
補足	本機能の処理には、状況によって最大30秒程度掛かる。 serverには、IPv4アドレス("x.x.x.x"形式)またはホスト名を指定することができ る。	

【使い方の例】

```
char *svr = "arduino.cc";
if (a3gim.connectTCP(svr, 80) == 0) {
    // GET Request for google site
    a3gim.write("GET / HTTP/1.0$n");
    a3gim.write("HOST:");
    a3gim.write(svr);
    a3gim.write("$n$n");
    // Get resopnse..
}
else {
    Serial.println("Error: can't connect");
}
```









6. TCP/IP機能の関数 disconnectTCP

int disconnectTCP(void)		
機能概要	接続しているTCPコネクションを切断する	
引数	なし	
戻り値	0:正常に切断できた時	
	0以外: エラーが発生した時(戻り値はエラー番号を表す)	
補足	本機能の処理には、状況によって1〜数秒程度掛かる。 本ライブラリの終了関数endでは、TCPコネクションは自動的に切断しない。その ため、必要に応じて必ず本関数を呼び出してTCPコネクションを明示的に切断する こと。	







6. TCP/IP機能の関数 getStatusTCP

int getStatusTCP(int *status,int *tcpnotif, logn *remainedBytes, long *receivedBytes)		
機能概要	接続しているTCPコネクションを切断する	
引数	status: TCPコネクションのステータス(別表①参照) tcpnotif: TCPコネクションで最後に発生したエラーの内容(別表②参照) remainedBytes: 相手に送信できていないデータのサイズ(バイト数) receivedBytes: 受信したがまだread()していないデータのサイズ(バイト数)	
戻り値	0:正常に切断できた時	
大ツ旭	0以外 : エラーが発生した時(戻り値はエラー番号を表す)	
補足	本機能の処理には、状況によって1〜数秒程度掛かる。 本ライブラリの終了関数endでは、TCPコネクションは自動的に切断しない。その ため、必要に応じて必ず本関数を呼び出してTCPコネクションを明示的に切断する こと。	







6. TCP/IP機能の関数 getStatusTCP

表① statusの意味

status	意味
0	未接続
1	接続済み
2	接続失敗
3	クローズした
4	接続中
5	アイドルタイム*2のカウント開始
6	アイドルタイム*2のカウント取り消し

^{*2} TCPコネクションが切断された時からアイドルタイムのカウントが開始される。アイドルタイムが30秒となった時に、3Gネットワークのセッションが解放される。アイドルタイム中にread()やwrite()を行うと、アイドルタイムはいったんリセットされる。

【エラー発生時の対処方法】

何らかのエラーが発生した場合は、通常、 TCPコネクションを切断して、再度接続からやり直す ことを推奨する。

表② tcpnotifの意味

tcpnotif	意味
0	ネットワークエラー
1	ソケットエラー
2	メモリ問題
3	DNS問題
4	TCPが相手から切断された
5	TCPコネクションエラー
6	一般的なエラー
7	クライアントからのリクエスト受付けエラー*1
8	AT+KTCPSNDで文字待ちが発生*1
9	セッションIDがおかしい*1
10	セッションは使用中である
11	すべてのセッションは使用中である*1

^{*1} 通常は発生しない内部エラー







6. TCP/IP機能の関数 writeBegin

int writeBegin(size_t sz)		
機能概要	指定したサイズszのデータをTCPコネクションに対して書き込む準備をする	
引数	sz:データのサイズ(バイト数)、最大 a3gsMAX_TUNNEL_DATA_LENGTH	
戻り値	0:正常に準備ができた時	
	-1: エラーが発生した時(引数szがおかしい)	
補足	本機能の処理には、状況によって数秒程度掛かる。 バイナリデータを相手へ書き込むための最速の関数である。\$エスケープが不要であり、一度の書き込めるサイズも大きいため、write()関数よりも高速である。 本関数を呼び出した後は、指定したサイズ分のデータ(バイナリデータもそのままでOK)を、シリアルa3gSerialに対して直接書き込む。正確にszバイト分のデータを書き込む必要がある。szに満たない場合は、30秒のタイムアウト後に、szバイトに満たない不足分のデータとして、3GIMが自動的に0x00バイトを充当する。通信速度を115,200bpsに設定している場合は、フロー制御を行っていないため1024バイトのデータ送るたびに5ミリ秒以上のディレイを挿入する必要がある。	

【使い方の例】

```
char *svr = "someserver.aaa";
uint8_t data[256];
if (a3gim.connectTCP(svr, 80) == 0) {
    a3gim.writeBegin(sizeof(data));
    for (int i = 0; i < sizeof(data); i++)
        a3gSerial.write(data[i]);
    int len = sizeof(buf) - 1;
    char buf[20];
    if (a3gs.getResult(buf, &len, 60000) != 0)
        // Error handling ..
}</pre>
```

左記の使い方にあるように、writeBegin()を呼んだ後は、シリアルa3gSerialに対して直接write()やprint()を使って所定のサイズ分のデータを書き込む。

書き込んだ後は、getResult()関数を呼び出して、書き込みの成否をチェックする。

All Rights Reserved.

Open Wireless Alliance





6. TCP/IP機能の関数 read

● バリエーション1 (バイナリデータの読み出し可)

int read(void)		
機能概要	現在のTCPコネクションから 1 バイトのデータを読み出す	
引数※	なし	
	0~0xFF : 読み出した1バイトのデータ	
戻り値	-1 : エラーが発生した時(コネクションがcloseされた、エラーが発生した、 タイムアウトした)	
	-2 : データが読み出せなかった時 (データがない時)	
補足	本関数は常にノンブロッキングで動作する。そのため、読み出せるデータがない時は直ちに呼び出し元ヘリターンする。 本関数は、コネクションがcloseされた時やエラーが発生した時は、直ちに呼び出し元ヘリターンする。	

【使い方の例】

```
char *svr = "arduino.cc";
if (a3gim.connectTCP(svr, 80) == 0) {
    // GET Request for google site
    a3gim.write("GET / HTTP/1.0$n");
    a3gim.write("HOST:");
    a3gim.write(svr);
    a3gim.write("$n$n");
    handleResponse();
}
else
Serial.println("Error: can't connect");
```

```
void handleResponse(void) {
int c;
while ((c = a3gim.read()) > 0) {
    // 読み出した文字cを処理する
    }
}
```

16 3G alliance All Rights Reserved.









6. TCP/IP機能の関数 read

● バリエーション2 (テキストデータの読み出しのみ)

int read(char* result, int resultlength)		
機能概要	現在のTCPコネクションから最大resultlengthバイトのデータを読み出す	
引数	result:[OUT] 読み出したデータを格納するバッファアドレス resultlength:呼び出し側で確保したバッファのサイズ(バイト数)	
	1~(resultlength-1): 正常に読み出した時(読み出したバイト数を返す)	
戻り値	0: データが読み出せなかった時	
	0未満:エラーが発生した時(コネクションがcloseされた、エラーが発生した)	
補足	本関数は常にノンブロッキングで動作する。そのため、読み出せるデータがない時は直ちに呼び出し元ヘリターンする。本関数は、コネクションがcloseされた時やエラーが発生した時は、直ちに呼び出し元ヘリターンする。resultで返却するデータは、ヌル文字('¥0')で終端させる。	

```
【使い方の例】
```

```
void handleResponse(void) {
  char res[a3gimMAX_RESULT_LENGTH+1];
  int nbytes;
  while(a3gim.read(res,a3gimMAX_RESULT_LENGTH+1) > 0) {
    Serial.print(res);
  }
}
```









6. TCP/IP機能の関数 read

● バリエーション3 (バイナリデータの読み出し可)

int read(uint8_t* buffer, size_t sz)		
機能概要	現在のTCPコネクションから最大szバイトのデータを読み出す	
引数	buffer :[OUT] 読み出したデータを格納するバッファアドレス sz :呼び出し側で確保したバッファbufferのサイズ(バイト数)	
	1~sz:正常に読み出した時(読み出したバイト数を返す)	
戻り値	0 : データが読み出せなかった時	
	0未満 :エラーが発生した時(コネクションがcloseされた、エラーが発生した)	
補足	本関数は常にノンブロッキングで動作する。そのため、読み出せるデータがない時は直ちに呼び出し元へリターンする。本関数は、コネクションがcloseされた時やエラーが発生した時は、直ちに呼び出し元へリターンする。機能はバリエーション2と同じであるが、バイナリデータを扱う場合は本関数を使用すること。バリエーション2と異なり、読み出したデータをヌル文字('¥0')で終端することはしない。	







6. TCP/IP機能の関数 write

● バリエーション1 (バイナリデータの書き出し可能)

int write(uint8_t c)	
機能概要	現在のTCPコネクションへ1バイトのデータを書き出す
引数	c:書き出すデータ
	1:正常に書き出した時
戻り値	0未満 :エラーが発生した時(コネクションがcloseされた、エラーが発生した、 タイムアウトした)
補足	本関数の処理には、状況によって最大30秒程度掛かる データcとして、制御文字、ヌル文字、あるいは特殊文字(ダブルクォート、\$)等も そのまま指定することができる。 本関数は同期処理であるため、コネクションへデータを書き出すまで、コネクションがcloseされるまで、エラーが発生するまで、あるいはタイムアウトするまで呼び 出し元に制御は戻らない。







6. TCP/IP機能の関数 write

● バリエーション2 (テキストデータの書き出しのみ)

int write(const char* str)	
機能概要	現在のTCPコネクションへ文字列データを書き出す
引数	str:書き出す文字列データ('¥0'で終端する文字配列)
	1以上:正常に書き出した時(書き出したバイト数を返す)
戻り値	0未満 :エラーが発生した時(コネクションがcloseされた、エラーが発生した、 タイムアウトした)
補足	本関数の処理には、状況によって最大30秒程度掛かる。 本関数は同期処理であるため、コネクションへデータを書き出すまで、コネクションがcloseされるまで、エラーが発生するまで、あるいはタイムアウトするまで呼び出し元に制御は戻らない。 バリエーション1・3と異なり、バイナリデータを取り扱うためのエスケープ処理を実行しないため、これらに比べて高速に実行できる。

【使い方の例】

```
char *svr = "arduino.cc";
if (a3gim.connectTCP(svr, 80) == 0) {
    // GET Request for google site
    a3gim.write("GET / HTTP/1.0$n");
    a3gim.write("HOST:");
    a3gim.write(svr);
    a3gim.write("$n$n");
    // Get resopnse..
}
```

12-2016 3G alliance All Rights Reserved.









6. TCP/IP機能の関数 write

● バリエーション3 (バイナリデータの書き出し可能)

int write(const uint8_t* buffer, size_t sz)		
機能概要	現在のTCPコネクションへ指定したバイト数のデータを書き出す	
引数	buffer :書き出すデータ(バイト配列) sz :データのサイズ(バイト数)	
	1以上:正常に書き出した時(書き出したバイト数を返す)	
戻り値	0未満 :エラーが発生した時(コネクションがcloseされた、エラーが発生した、 タイムアウトした)	
補足	バリエーション2ではデータの中にヌル文字('¥0')を取り扱うことができないが、 本バリエーションではデータをエスケープ処理するため、データの中に制御文字、 ヌル文字、あるいは特殊文字(ダブルクォート、\$)をそのまま含めることができる。 本関数の処理には、状況によって最大30秒程度掛かる。 本関数は同期処理であるため、コネクションへデータを書き出すまで、コネク ションがcloseされるまで、エラーが発生するまで、あるいはタイムアウトするま で呼び出し元に制御は戻らない。	

【使い方の例】

```
char *svr = "192.168.1.1";
uint8_t binaryData[] = { 0x0, 0x1, 0x2, 0x3, ...};
if (a3gim.connectTCP(svr, 8080) == 0) {
   a3gim.write(binaryData, sizeof(binaryData));
}
```









7. プロファイルの関数

提供関数ライブラリ

分類	メソッド名	機能概要	補足
プロファイル	setDefaultProfile	デフォルトプロファイルを設定	
	getDefaultProfile	デフォルトプロファイルを取得	

【注意事項】プロファイルは、通信サービス事業体が提供するSIMカードを利用するための設定情報である。 詳細は、setDefaultProfile 関数の説明を参照。





7. プロファイルの関数 setDefaultProfile

int setDefaultProfile(const char *apn, const char *user, const car *password)		
機能概要	デフォルトのプロファイルを設定する	
引数	apn : SIMカードのAPN情報 user : SIMカードのユーザ名 password : SIMカードのパスワード	
戻り値	0:正常に設定できた時	
	0以外:設定できなかった時(引数の指定が間違っている等)	
補足	設定したデフォルトのプロファイル番号は、内臓マイコンのフラッシュROMに記録されるため、電源をOFFにしても維持される(再度、本ライブラリにて設定するまで有効である) 3 GIM V2.0の出荷時の設定プロファイル情報は、下記の通りである: 出荷時デフォルト: IIJ様の iijmio サービス(iijmio.jp/mio@iij/iij)	







7. プロファイルの関数 getDefaultProfile

int getDefaultProfile(char *apn, char *user, char *password)	
機能概要	デフォルトのプロファイル番号を取得する
引数	apn : SIMカードのAPN情報 user : SIMカードのユーザ名 password : SIMカードのパスワード
戻り値	0:正常に取得できた時
	0以外 : 取得できなかった時
補足	

【使い方の例】

```
int pfNum;
if (a3gim.getDefaultProfile(&pfNum) == 0) {
    Serial.print("Default Profile No is ");
    switch (pfNum == 1)
    case 1:
        Serial.println("docomo mopera");
    case 2:
        Serial.println("IIJmio");
    case 3:
        Serial.println("IIJmobile");
```

```
case 11:
Serial.println("bmobile");
case 15:
Serial.println("DTI ServerMan");
default:
Serial.println("unknown profile");
}
```





8. ATコマンド稼働

int enterAT(unit32_t duration)		
機能概要	ATコマンドパススルーモードに切り替え	
duration: ATコマンドパススルーモードから戻るまでの時間(単位:0.5 引数		
戻り値	0:ATコマンドモードに切り替わった場合	
大ツ胆	1:引数の指定がおかしい時など(モードは切り替わらない)	
補足	ATコマンドの使用に制限はないため、HL8548-Gの設定を任意に変更することができる。しかし、変更した設定等によっては、gw3gファームウェアの動作に支障をきたす場合があるので、十分留意すること。 利用できるATコマンドの詳細は、HL8548-Gの開発元であるSierra Wireless社のサイトで公開されている「AT Commands Interface Guide - AirPrime HL6 and HL8 Series」を参照のこと。 ※ATコマンドそのものに関しては、Tabrainでは技術的なサポートは致しません。(ユーザの責任範囲でご利用ください)	



1.	a3gim.zip ファイルの内容	p.61
2.	a3gim.zip ファイルの概要	p.62
3.	a3gimイントール完了の確認	p.63

IV編 a3gim のインストール





1. a3gim.zip ファイル

NPO法人オープンワイヤレスアライアンスから、ライブラリおよびサンプルスケッチ内容が、a3gim.zip(圧縮)ファイルとして提供されます。 (随時バージョンアップ版を提供

して参ります)

http://tabrain.jp/3GIM_V2.0/a3gim.zip

□ 名前	サイズ	ファイルの種類	
examples		ファイル フォルダ	
a3gim.cpp	47,497	C++ Source File	١
a3gim.h	9,020	H ファイル	
keywords.txt	2,078	テキスト ドキュメント	
READ_ME.txt	2,705	テキスト ドキュメント	

このzipファイルを解凍し、Arduino のIDE(統合化開発環境)のファイル群にある 「libraries」の配下に複写(移動)をしてください。

Arduino IDE フォルダ	フォルダ		
	drivers		
	examples		
最新版ArduinoIDE	hardware		
arduino-1.7.8-windows	java		
arduino-1.6.7-windows 配下	lib		1
	libraries	※ こちらの配下に複写してください	•
	reference		
	tools		

01.Basics 02.Digital 03.Analog 04.Communication 05.Control 06.Sensors 07.Display 08.Strings 09.USB 10.StarterKit_BasicKit 11.ArduinoISP Examples from Libraries Adafruit DotStar Adafruit Motor Shield V2 Library Adafruit NeoPixel Bridge CurieBle CurieImu CurieSoftwareSerial CurieTime Ethernet Firmata GSM LiquidCrystal OneWire SD Servo SPI Stepper Temboo Wire RETIRED Examples from Custom Libraries a3gim

Built-in Examples





2. a3gim.zip ファイルの概要

・ 提供される a3gim.zip には、以下のファイル群が含まれています。鍵括弧内 【ページ】 は、V編のサンプル・スケッチのページです。

	フォルダ/ファイル(スケッチ)	概要
	¥blink_led¥blink_led.ino [p.66]	LEDを点滅させるスケッチ
	¥check_baudrate¥check_baudrate.ino [p.67]	IEM通信モジュールの通信ボーレートを確認するスケッチ
	¥check_rssi¥check_rssi.ino [p.68]	電波状態を確認するスケッチ
	¥check_service¥check_service.ino [p.69]	利用できるネットワークサービスを確認するスケッチ
	¥get_imei¥get_imei.ino【p.70】	IEMモジュールのIDを取得するスケッチ
	¥get_location¥get_location.ino [p.71]	現在位置を取得するスケッチ■
	¥get_location_agps¥get_location_apgs.ino【p.72】	現在位置をAssisted GPSによって取得するスケッチ※■
	¥get_status¥get_status.ino [p.73]	getStatusを呼び出すスケッチ
example 【事例】	¥get_time¥get_time.ino [p.74]	日付・日時を取り出すスケッチ <mark>※</mark>
13 1/32	¥get_time2¥get_time2.ino【p.75】	'70年1月1日からの総合秒数を取り出すスケッチ※
	¥http_get¥http_get.ino [p.76]	httpGETを使うサンプルスケッチ <mark>※</mark>
	¥sample_TCPIP¥sample_TCPIP.ino【p.77】	TCP/IP機能を利用するサンプルスケッチ※
	¥send_sms¥send_sms.ino [p.79]	ショートメッセージを送信するスケッチ※★
	¥set_baudrate¥set_baudrate.ino [p.80]	通信速度を変更するスケッチ <mark><注意></mark>
	¥set_defaultprofie ¥set_defaultprofie.ino [p.81]	デフォルトのプロファイル(APN)を切り替えるスケッチ
	¥tweet_sample ¥tweet_sample.ino [p.82]	Tweetを活用したスケッチ※
	¥xively_sample¥xively_sample.ino [p.83]	無償クラウドxivelyを利用したスケッチ <mark>※</mark>
a3gim.cpp	"a3gim" 実装ファイル	
a3gim.h	"a3gim" ヘッダーファイル	
keywords.txt	a3gimライブラリのキーワード定義	
READ_ME.txt	a3gim R4.0 リリースノート	

※印のついたスケッチは、SIMカードが必要。★印のついたスケッチは、SMS対応SIMカードが必要。■印のついたスケッチはGPSアンテナ必要。 <注意>IEM通信モジュールとArduinoとの通信速度の設定で、内容理解の上利用。



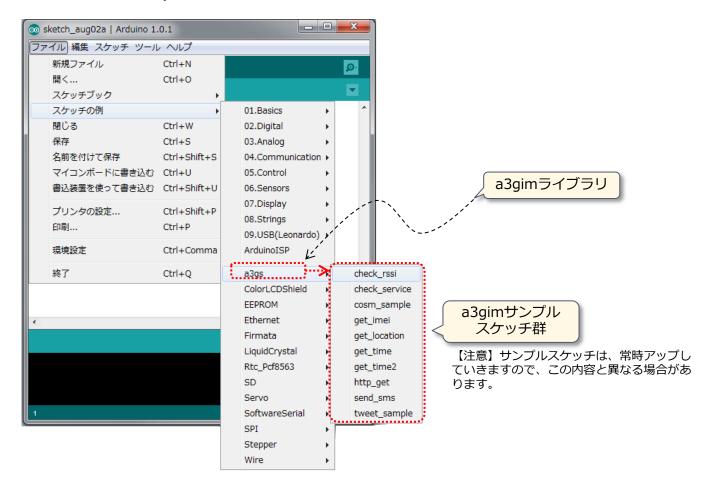






3. a3gim インストール完了の確認

a3gimのライブラリおよびサンプルスケッチが正しくインストールされた場合には、Arduino IDE環境下のプルダウンメニュー「ファイル(File)」の「スケッチの例」において、以下のように画面表示されます。





V編 サンプル・スケッチ

【注意事項】本稿で扱うスケッチ(プログラム)は、IDEバージョン(arduino.org版) Arduino 1.7.8上で作成・確認したものです。その他のバージョンでは稼働しない場合があります。





1. blink_led.ino [LEDの点滅]

本スケッチは、3Gシールド基板上に搭載れているLED1のON/OFF

を行う事例です。

```
// 3GIM(V2) sample sketch -- setLED
#include <SoftwareSerial.h>
#include <a3gim.h>
```

【シリアルモニタ表示例】

Ready. Initializing.. Succeeded. Blinking..



```
#define INTERVAL 1000 // Blink interval
#define baudrate
                                9600UI
const int powerPin = 7; // 3gim power pinIf not using power control, 0 is set.
void setup()
 Serial.begin(baudrate);
 delay(3000); // Wait for Start Serial Monitor
 Serial.println("Ready.");
 retry:
 Serial.print("Initializing.. ");
 if (a3gs.start(powerPin) == 0 \&\& a3gs.begin(0, baudrate) == 0)
  Serial.println("Succeeded.");
 else {
  Serial.println("Failed.");
  Serial.println("Shutdown..");
  a3qs.end();
                                            初期化失敗時は、
  a3gs.shutdown();
                                              リトライする
  delay(30000);
  goto _retry; // Repeat
 Serial.println("Blinking..");
void loop()
 a3qs.setLED(true);
 delay(INTERVAL);
                                     LEDのOn/Off処理
 a3qs.setLED(false);
 delay(INTERVAL);
// END
```







2. check_baudrate.ino [通信ボーレート確認のサンプル]

本スケッチは、IEMモジュールの通信ボーレートの確認を行うもので、9600から115200までの間のどれに設定されているかを確認していきます。

【シリアルモニタ表示例】

Ready.
Initializing..
Try baudrate: 9600
Recognize succeeded.
Current baudrate is 9600 bps.



```
void setup()
 Serial.begin(9600);
 Serial.println("Ready.");
 Serial.println("Initializing.. ");
 for (int i = 0; i < 7; i++) {
  if (a3qs.start(powerPin) == 0) {
    Serial.print("Try baudrate: ");
    Serial.println(baudrates[i]);
    if (a3gs.begin(0, baudrates[i]) == 0) {
     Serial.println("Recognize succeeded.");
     Serial.print("Current baudrate is ");
     Serial.print(baudrates[i]);
     Serial.println(" bps.");
     return;
    Serial.println("Failed.");
    a3qs.end();
    a3gs.shutdown();
 Serial.println("Can't recognize baudrate.");
void loop()
// END
```







3. check_rssi.ino 【受信信号(電波)強度取得】

Arduino上の3Gシールドが、電波強度「rssi」について、3G電波を捉えているかを、数値によって返します。

【サンプルスケッチ(check_rssi.ino)】

```
// 3GIM(V2) sample skech -- getRSSI

#include <SoftwareSerial.h>
#include "a3gim.h"

#define baudrate 9600UL
const int powerPin = 7; // 3gim power pin(If not using power control, 0 is set.)
```

【シリアルモニタ表示例】



```
void setup()
 Serial.begin(baudrate);
 delay(3000); // Wait for Start Serial Monitor
 Serial.println("Ready.");
 Serial.print("Initializing.. ");
 if (a3gs.start(powerPin) == 0 \&\& a3gs.begin(0, baudrate) == 0) {
  Serial.println("Succeeded.");
   int rssi;
  if (a3gs.getRSSI(rssi) == 0) {
    Serial.print("RSSI = ");
    Serial.print(rssi);
    Serial.println(" dBm");
                                         電波強度を引数rssiに返す
 else
  Serial.println("Failed.");
 Serial.println("Shutdown..");
 a3qs.end();
 a3gs.shutdown();
void loop()
// END
```

「-110」に近い 場合は、アンテ ナが外れている かもしれません。









4. check_service.ino [SIMカードサービス形態取得]

SIMカードのサービス形態を取得し表示するスケッチです。

【【サンプルスケッチ(check_sercvce.ino)】

```
// 3GIM(V2) sample sketch -- getServices

#include <SoftwareSerial.h>
#include "a3gim.h"

#define baudrate 9600UL
const int powerPin = 7;
// 3gim power pin(If not using power control, 0 is set.)

void setup()
{
    Serial.begin(baudrate);
    delay(3000); // Wait for Start Serial Monitor
    Serial.println("Ready.");

    Serial.print("Initializing.. ");
```

【シリアルモニタ表示例】

Ready.
Initializing.. Succeeded.
Packet Service Only.
Shutdown..



```
a3gsSRV_NO(= 0): サービス利用不可
a3gsSRV_PS(= 1): パケット通信サービスのみ
a3gsSRV_CS(= 2): 音声通信(+SMS)サービスのみ
a3gsSRV_BOTH(= 3): パケット通信、音声通信(SMS)いずれも
```

```
if (a3gs.start(powerPin) == 0 && a3gs.begin(0, baudrate) == 0) {
  Serial.println("Succeeded.");
  int status;
  if (a3gs.getServices(status) == 0) {
    switch (status) {
     case a3qsSRV NO:
      Serial.println("No Service.");
       break;
     case a3qsSRV PS:
      Serial.println("Packet Service Only.");
       break:
     case a3gsSRV CS:
                                                     // Never returned at R4.0
      Serial.println("Voice Service Only."); // also SMS
     case a3gsSRV BOTH: // Never returned at R4.0
      Serial.println("Packet And Voice Services.");
       break:
     default:
       break;
 else
  Serial.println("Failed.");
 Serial.println("Shutdown..");
 a3qs.end();
 a3gs.shutdown();
void loop()
// END
```









5. get_imei.ino 【IEMモジュール携帯端末識別番号取得】

▶ IEMモジュールの固有ID(IMEI:携帯端末識別番号)を取得し表示するスケッチです。

【サンプルスケッチ(get_time.ino)】

```
// 3GIM(V2) sample sketch -- getIMEI
#include <SoftwareSerial.h>
#include "a3gim.h"
#define baudrate 9600UL
const int powerPin = 7; // 3gim power pin(If not using power control, 0 is set.)
void setup()
 Serial.begin(baudrate);
 delay(3000); // Wait for Start Serial Monitor
 Serial.println("Ready.");
 Serial.print("Initializing.. ");
 if (a3gs.start(powerPin) == 0 && a3gs.begin(0, baudrate) == 0) {
  Serial.println("Succeeded.");
   char imei[a3gsIMEI_SIZE];
  if (a3gs.getIMEI(imei) == 0) {
    Serial.print("IMEI: ");
    Serial.println(imei);
                              IMEI取得関数
 else
   Serial.println("Failed.");
 Serial.println("Shutdown..");
 a3qs.end();
 a3gs.shutdown();
void loop()
}
// END
```

【シリアルモニタ表示例】



Ready. Initializing... <u>Succeeded.</u> IMEI: <u>359516050164625</u> Shutdown...

IMEI
(携带端末識別番号)







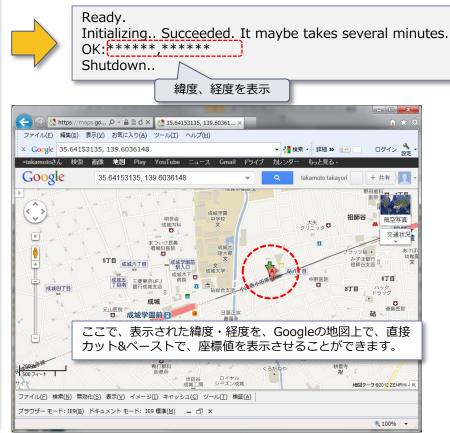
6. get_location.ino 【現在位置取得】

現在位置情報をGPSより取得表示するサンプルです。

```
// 3GIM(V2) sample sketch -- getLocation
#include <SoftwareSerial.h>
#include "a3gim.h"
#define baudrate 9600UL
const int powerPin = 7; // 3gim power pin(If not using power control, 0 is set.)
void setup()
 Serial.begin(baudrate);
 delay(3000); // Wait for start serial monitor
 Serial.println("Ready.");
 Serial.print("Initializing.. ");
 if (a3gs.start(powerPin) == 0 \&\& a3gs.begin(0, baudrate) == 0) {
  Serial.println("Succeeded. It maybe takes several minutes.");
  char lat[15], lng[15];
  if (a3gs.getLocation(a3gsMPBASED, lat, lng) == 0) {
    Serial.print("OK: ");
    Serial.print(lat);
                                        位置情報取得
    Serial.print(", ");
    Serial.println(lng);
    Serial.println("Sorry, I don't know this location.");
  Serial.println("Failed.");
 Serial.println("Shutdown..");
 a3gs.end();
 a3gs.shutdown();
void loop()
// END
```

【注意事項】 GPS機能は、アンテナ特性が影響しますので、アンテナの感度の良いもので、なるべく周りの障害物を避け、さらに周辺にノイズ発生源が無いなどの条件で、位置情報が取得できます。

【シリアルモニタ表示例】











7. get_location2.ino 【現在位置取得】

・ 現在位置情報をAssisted GPSより取得表示するサンプルです。(こちらは3GとGPS電波状態が良い場所だと30 ・ 秒以内で位置情報が取得できます)

```
// 3GIM(V2) sample sketch -- getLocation2
#include <SoftwareSerial.h>
#include "a3gim.h"
#define baudrate 9600UL
const int powerPin = 7; // 3qim power pin(If not using power control, 0 is set.)
void setup()
 Serial.begin(baudrate);
 delay(3000); // Wait for start serial monitor
 Serial.println("Ready.");
 Serial.print("Initializing.. ");
 if (a3gs.start(powerPin) == 0 && a3gs.begin(0, baudrate) == 0) {
  Serial.println("Succeeded.");
  setupAGPS();
  char lat[15], lng[15], utc[7], height[8];
   int quality, number;
  if (a3gs.getLocation2(lat, lng, height, utc, &quality, &number) == 0) {
    Serial.print("OK: ");
    Serial.print(lat); Serial.print(","); Serial.print(lng); Serial.print(",");
                                                  位置情報取得
                           Serial.print(",");
    Serial.print(height);
                         Serial.print(",");
    Serial.print(utc);
                             Serial.print(",");
    Serial.print(quality);
    Serial.println(number);
          Serial.println("Sorry, I don't know this location.");
  else
 else
       Serial.println("Failed.");
 Serial.println("Shutdown..");
 a3qs.end();
 a3gs.shutdown();
void loop() {}
```

```
// setup AGPS function
void setupAGPS()
{
  char apn[20], user[20], password[20];
  if (a3gs.getDefaultProfile(apn, user, password) == 0) {
    char atwppp[50];
    sprintf(atwppp,"at+wppp=2,4,\fomale*"\sets\fomale*",\fomale*"\sets\fomale*",user,password);
    Serial.println(atwppp);
    a3gs.enterAT(2);
    a3gSerial.println(atwppp);
    delay(200);
    Serial.println("Assisted GPS set OK");
  }
  else    Serial.println("NG: getDefaultProfile(), can't use AGPS..");
}
// END
```

【注意事項】 本機能は、予めSIMカードが正しく設定されていることが前提条件となります。

【シリアルモニタ表示例】

Ready.



```
Initializing.. Succeeded.
at+wppp=2,4,"-","-"
Assisted GPS set OK
OK: 35.641879,139.604130,32.7,083151,1,6
Shutdown..
```

緯度、経度、高度、時間、精度、衛星数を表示

ここで、表示された緯度・経度を、Googleの地図上で、直接カット&ペーストで、座標値を表示させることができます。









8. get_status.ino 【現在の通信状態取得】

3G通信モジュールの通信状態を返します。

```
// 3GIM(V2) sample sketch -- getStatus

#include <SoftwareSerial.h>
#include <a3gim.h>

#define baudrate 9600UL
const int powerPin = 7; // 3gim power pin(If not using power control, 0 is set.)
```

【シリアルモニタ表示例】

Ready.
Initializing.. Succeeded.
Status is IDLE
Shutdown..



状態:

ERROR: エラー状態 IDLE: 動いていない状態

READY: 待機状態

TCPCONNECTEDCLINENT: TCP接続状態

Unknown: 不明

```
void setup()
 Serial.begin(baudrate);
 delay(3000); // Wait for Start Serial Monitor
 Serial.println("Ready."):
 Serial.print("Initializing.. ");
 if (a3gs.start(powerPin) == 0 \&\& a3gs.begin(0, baudrate) == 0) {
  Serial.println("Succeeded.");
  int status:
  status = a3gs.getStatus();
  Serial.print("Status is ");
  switch (status) {
   case A3GS::ERROR:
     Serial.println("ERROR"):
     break:
    case A3GS::IDLE:
     Serial.println("IDLE"):
     break:
    case A3GS::READY:
     Serial.println("READY");
     break:
    case A3GS::TCPCONNECTEDCLIENT:
     Serial.println("TCPCONNECTEDCLIENT");
     break:
    default:
     Serial.println("Unknown");
     break:
 Serial.println("Shutdown..");
 a3gs.end();
 a3gs.shutdown();
void loop() {}
// END
```









9. get_time.ino [時刻取得1]

▶ IEMモジュールにあるタイマー機能を使って、年月日および時刻を取得するスケッチです。

【サンプルスケッチ(get_time.ino)】

```
// 3GIM(V2) sample sketch -- getTime
#include <SoftwareSerial.h>
#include "a3gim.h"
#define baudrate 9600UL
const int powerPin = 7; // 3qim power pin(If not using power control, 0 is set.)
void setup()
 Serial.begin(baudrate);
 delay(3000); // Wait for Start Serial Monitor
 Serial.println("Ready.");
 Serial.print("Initializing.. ");
 if (a3gs.start(powerPin) == 0 \&\& a3gs.begin(0, baudrate) == 0) {
  Serial.println("Succeeded.");
  char date[a3gsDATE_SIZE], time[a3gsTIME_SIZE];
   if (a3gs.getTime(date, time) == 0) {
   Serial.print(date);
                                 時刻取得
    Serial.print(" ");
    Serial.println(time);
 else
  Serial.println("Failed.");
 Serial.println("Shutdown..");
 a3qs.end();
 a3gs.shutdown();
void loop()
// END
```

【シリアルモニタ表示例】



Ready. Initializing.. Succeeded. 2012/10/05 07:29:00 Shutdown..

【注意事項】 現在時刻の取得は、IEMモジュールの継続運用によって、設定されますので、初期状態では、1980年1月5日16:00の時刻か、日本時間の16時間遅れの時刻で表示される場合があります。利用時間経過とともに、日本時間に自動調整が行われます。









10. get_time2.ino [時刻取得2]

IEMモジュールにあるタイマー機能を使って、1970年1月1日からの総数秒を取得するスケッチです。

【サンプルスケッチ(get time2.ino)】

```
// 3GIM(V2) sample sketch -- getTime2
// check result by http://www.epochconverter.com/
#include <SoftwareSerial.h>
#include "a3gim.h"
#define baudrate 9600UL
const int powerPin = 7; // 3gim power pin(If not using power control, 0 is set.)
void setup()
 Serial.begin(baudrate);
 delay(3000); // Wait for Start Serial Monitor
 Serial.println("Ready.");
 Serial.print("Initializing.. ");
 if (a3gs.start(powerPin) == 0 && a3gs.begin(0, baudrate) == 0) {
   Serial.println("Succeeded.");
   uint32 t seconds:
   if (a3gs.getTime2(seconds) == 0) {
    Serial.print(seconds);
    Serial.println(" Sec.");
                                      時刻取得
   else
    Serial.println("Can't get seconds.");
 else
   Serial.println("Failed.");
 Serial.println("Shutdown..");
 a3as.end():
 a3gs.shutdown();
void loop()
// END
```

【シリアルモニタ表示例】



Ready. Initializing.. Succeeded. 1344179843 Sec. Shutdown...

【注意事項】 現在時刻の取得は、IEMモジュール の継続運用によって、設定されますので、初期状 態では、1980年1月5日16:00での経過時間か、 日本時間の16時間遅れの経過時間で表示される場 合があります。利用時間経過とともに、日本時間 の経過時間に自動調整が行われます。







11. http_get.ino 【サーバのレスポンス返却】

本スケッチでは、google サーバにアクセスし、レスポンスの格納先を表示させています。

※V1.1から、httpsにも対応しています(P.30参照)

【サンプルスケッチ(http_get.ino)】

```
#include <SoftwareSerial.h>
#include "a3gim.h"

#define baudrate 9600UL
const int powerPin = 7;
// 3gim power pin(If not using power control, 0 is set.)
const char *server = "www.arduino.org";
const char *path = "";
int port = 80;

char res[a3gsMAX_RESULT_LENGTH+1];
int len;
```

【シリアルモニタ表示例】

```
Ready.
Initializing.. Succeeded.
httpGET() requesting.. OK!
[<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title>Arduino - HomePage </title>
link rel="shortcut icon" type="image/x-icon"
href="http://arduino.cc/en/favicon]
Shutdown..
```

```
void setup()
 Serial.begin(baudrate);
 delay(3000); // Wait for Start Serial Monitor
 Serial.println("Ready.");
 Serial.print("Initializing.. ");
 if (a3gs.start(powerPin) == 0 && a3gs.begin(0, baudrate) == 0) {
  Serial.println("Succeeded.");
  Serial.print("httpGET() requesting.. ");
  len = sizeof(res):
  if (a3gs.httpGET(server, port, path, res, len) == 0) {
    Serial.println("OK!");
    Serial.print("[");
    Serial.print(res);
    Serial.println("]");
    Serial.print("Can't get HTTP response from ");
    Serial.println(server);
  Serial.println("Failed.");
 Serial.println("Shutdown..");
 a3gs.end();
 a3gs.shutdown();
void loop()
// END
```







12. sample_TCPIP.ino ① [TCP/IP機能のサンプル]

本スケッチは、TCP/IP機能を使ってhttp通信を行う事例です。 この例では、Arduinoの公式ページから、titleタグの中身を抜き出して、シリアルへ出力します。

改行文字等は、\$文字を使ったエスケープシーケンスを使って 記述します。

```
// 3GIM(V2) sample sketch -- connectTCP/disconnectTCP/read/write
// A title is extracted from a homepage.
#include <SoftwareSerial.h>
#include <a3qim.h>
#define baudrate 9600UL
const int powerPin = 7; // 3gim power pin(If not using power control,
0 is set.)
const char *server = "www.arduino.cc"; // URL to extract a title
const int port = 80:
char res[200];
boolean getTitle(char *res);
void setup()
 Serial.begin(baudrate);
 delay(3000); // Wait for Start Serial Monitor
 Serial.println("Ready.");
 redo:
 Serial.print("Initializing.. ");
 if (a3gs.start(powerPin) == 0 && a3gs.begin(0, baudrate) == 0) {
  Serial println("Succeeded ");
 if (a3gs.connectTCP(server, port) != 0) {
   Serial.println("connectTCP() can't connect");
   goto _end;
```

```
HTTP/GETリクエスト
// Send GET request
  a3gs.write("GET / HTTP/1.0$r$n");
  a3gs.write("HOST:");
  a3qs.write(server);
  a3qs.write("$r$n$r$n");
  // Receive response
  do {
   int nbytes;
  if ((nbytes = a3gs.read(res, sizeof(res)-1)) < 0) {
     Serial.println("read() failed");
     break;
  } while (! getTitle(res));
 else
  Serial.println("Failed.");
 Serial.println("Shutdown..");
 a3qs.end();
 a3qs.shutdown();
                                          Titleタグが取得できるまで、
                                            レスポンスを読み続ける
 delay(15000);
 goto redo; // Repeat
 end:
 while (1);
void loop() { }
```



TCP/IPの接続処理





12. sample_TCPIP.ino ② 【TCP/IP機能のサンプル】

```
boolean getTitle(char *p)
 char *title;
 while (*p!= '\u0') {
  if (*p++ != '<')
   continue; // skip not tag
  if (strncmp((const char *)p, "title>", 6))
    continue; // skip not title tag
  // title tag found
   p += 6;
   title = p;
                                                             レスポンスの中から
  while (*p!= '\(\frac{1}{2}\) \(\frac{1}{2}\) \(\frac{1}{2}\)
                                                           <title>タグを見つける
    p++;
   *p = '¥0';
   Serial.print(server);
   Serial.print(": Page title is \(\pm\)");
                                                       <title>タグの中身
   Serial.print(title);
                                                        だけを抽出する
   Serial.println("\(\text{""}\);
   return true;
 return false;
// END
```

【シリアルモニタ表示例】

Ready.

Initializing.. Succeeded.

www.arduino.cc : Page title is "Arduino -

HomePage "

Shutdown..









13. send_sms.ino (ショートメッセージ送信)

ショートメッセージを特定の携帯電話に送信するスケッチです。

```
// 3GIM(V2) sample sketch -- sendSMS (Japanease message version)
// PLEASE REPLACE "msn" WITH CORRECT TELEPHONE NUMBER BEFORE UPLOAD THIS SKETCH.
#include <SoftwareSerial.h>
#include "a3gim.h"

SMSを受信する電話番号を設定

#define baudrate 9600UL
const int powerPin = 7; // 3gim pin(If not using power control, 0 is set.)

char *msn = "09000000000" // Replace your phone number!
char *msg = "TEST MESSAGE. HELLO!"; // ASCII String
// Japanease(Multi-byte code) is not supported in 3GIM(V2)

void setup
{
Serial.be
```

【シリアルモニタ表示例】

Ready.
Initializing.. Succeeded.
SMS Sending.. OK!
Shutdown..



【モバイル側のSMS内容表示】

TEST MESSAGE. HELLO!

【注意1】SMS機能は、音声通話ができるSIMカードでないと働きません。

【注意 2】送信できる文字コードは、アスキーコードか、漢字コード(日本語)ではUNICODEとなっています。

「これはテストです」(UNICODE)

```
void setup()
 Serial.begin(baudrate);
 delay(3000); // Wait for Start Serial Monitor
 Serial.println("Ready.");
 Serial.print("Initializing.. ");
 if (a3gs.start(powerPin) == 0 \&\& a3gs.begin(0, baudrate) == 0) {
   Serial.println("Succeeded.");
  Serial.print("SMS Sending.. ");
   if (a3gs.sendSMS(msn, msg) == 0)
    Serial.println("OK!");
    Serial.println("Can't send SMS.");
 else
  Serial.println("Failed.");
 Serial.println("Shutdown..");
 a3qs.end();
 a3gs.shutdown();
void loop() { }
// END
```







14.set baudrate.ino (通信速度の変更)

通信速度(ボーレート)を変更します。

```
// 3GIM(V2) sample sketch -- setBaudrate
// You need to use "setBaudrate()" function CAREFULLY.
// Because, 3G Shield becomes impossible to use when setting up
// the value which is not suitable.
// This function is used when gathering the communication speed of
// Arruino and 3GIM Shield using HardwareSerial instead of SoftwareSerial.
// In order for you to use HardwareSerial, you need to correct "a3qs.h"
// and "a3gim.cpp" appropriately.
// When using an "a3gim" library after performing this sketch,
  begin() is called in the following arguments:
      begin(0, NEW BAUDRATE);
#include <SoftwareSerial.h>
#include "a3gim.h"
                                              新しい通信速度 (bps)
#define NEW BAUDRATE
                                  38400
                                              現在の通信速度 (bps)
                                   9600
#define CURRENT BAUDRATE
const int powerPin = 7; // 3gim power pin(If not using power control, 0 is set.)
```

【シリアルモニタ表示例】

Ready. Initializing.. Succeeded. Baudrate was changed as 38400 bps. Shutdown...



```
void setup()
 Serial.begin(CURRENT_BAUDRATE);
 delay(3000); // Wait for Start Serial Monitor
 Serial.println("Ready.");
 Serial.print("Initializing.. ");
 if (a3gs.start(powerPin) == 0 \&\& a3gs.begin(0,
CURRENT BAUDRATE) == 0) {
//@ if (a3gs.start() == 0 && a3gs.begin(0, NEW BAUDRATE) ==
0) {
  Serial.println("Succeeded.");
  if (a3gs.setBaudrate(NEW_BAUDRATE) == 0) {
   Serial.print("Baudrate was changed as ");
    Serial.print(NEW BAUDRATE);
    Serial.println("bps.");
  Serial.println("Failed, baudrate was not changed.");
 Serial.println("Shutdown..");
 a3qs.end();
 a3qs.shutdown();
void loop()
// END
```









15. set_defaultprofile.ino (プロファイル変更)

本スケッチは、SIMカードのプロファイルを書き替えるもので、NTTドコモ通信サービス(=1)とIIJ個人向け通信サービス(IIJmio=2)、IIJ mobile(=3)、b-mobile(=11)、DTI ServersMan(=14)の切替を行う事例となります。

```
// 3GIM(V2) sample sketch -- setDefaultProfile and getDefaultProfile

#include <SoftwareSerial.h>
#include "a3gim.h"

#define baudrate 9600UL
const int powerPin = 7; // 3gim power pin(If not using power control, 0 is set.)

// New profile
const char *newApn = "x.y";
const char *newUser = "u";
const char *newPassword = "p";
```

このサンプルスケッチは、トグル(交互)設定機能 となっています。

【注意】SIMカードの提供は、通信キャリア以外にも、MVNOなどによっても提供されています。このSIMカードの種類は、現在多く出回っていて、この3Gシールドで使えるSIMカードは、IIJの法人向けSIMカードやIIJmio個人向けSIMカードなどに限定しています。

必ずしもNTTドコモのSIMカードでも、一部の機能では稼働するとは限りません。

(NPO法人3Gシールドアライアンスでは、IIJmioなどの販売提供を行っていますので、別途ご連絡頂けますと、ご提供して参ります)

```
void setup()
 Serial.begin(baudrate);
 delay(3000); // Wait for Start Serial Monitor
 Serial.println("Ready.");
 Serial.print("Initializing.. ");
 if (a3gs.start(powerPin) == 0 && a3gs.begin(0, baudrate) == 0) {
   Serial.println("Succeeded.");
                  // Get current default profile
   char apn[20], user[20], password[20];
   if (a3gs.getDefaultProfile(apn, user, password) == 0) {
    Serial.print("Default Profile Number is ");
    Serial.print(apn):
    Serial.print(",");
                                          プロファイル取り出し
    Serial.print(user);
    Serial.print(",");
    Serial.println(password);
                  // Set new default profile
   if (a3gs.setDefaultProfile(newApn, newUser, newPassword) == 0)
    Serial.println("Succeed.");
                                        プロファイル切替え
   else
    Serial.println("Failed.");
 else
   Serial.println("Failed.");
 Serial.println("Shutdown..");
 a3qs.end();
 a3gs.shutdown();
void loop() { }
// END
```

Copyright(c) 2012-2016 3G alliance All kignts keserveu.









16. tweet_sample.ino [ツィートデータ送信]

tweetにつぶやくサンプルとなります。

あらかじめ、ツイッターのアカウントを取得し、その TOKENを使って、サンプルスケッチを動かします。

ツイッターのアカウントは、ツイッター

https://twitter.com/

より、行ってください。

TOKEN(トークン)は、以下のサイトから取得してください。詳細は添付にて紹介しています。

http://arduino-tweet.appspot.com/

(注意) ツイッターは、同じ内容を連続して送ると、エラーとなってしまいます。



※ Twtterトークンは、【添付資料】参照

```
// 3GIM(V2) sample sketch – tweet
#include <SoftwareSerial.h>
#include "a3gim.h"
#define baudrate 9600UL
const int powerPin = 7; // 3gim power pinIf not using power control, 0 is set.
const char *token = YOUR_TOKEN_HERE;
                                                        Twtter トークン※
const char *message = "TWEET MESSAGE HERE";
 //-- Note: can't tweet same message continuously.
void setup()
 Serial.begin(baudrate);
 delay(3000); // Wait for Start Serial Monitor
 Serial.println("Ready.");
 Serial.print("Initializing.. ");
 if (a3gs.start(powerPin) == 0 \&\& a3gs.begin(0, baudrate) == 0) {
  Serial.println("Succeeded.");
  Serial.print("tweet() requesting.. ");
  if (a3gs.tweet(token, message) == 0)
    Serial.println("OK!");
  else
    Serial.println("Can't tweet.");
 else
  Serial.println("Failed.");
 Serial.println("Shutdown..");
 a3qs.end();
 a3gs.shutdown();
void loop()
// END
```



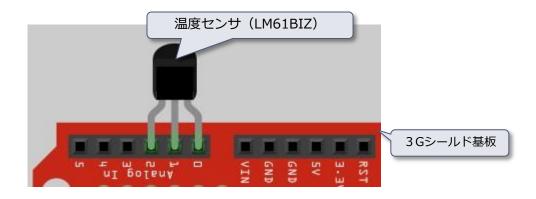






17. xively_sample.ino① 【クラウドとの連携】

- 無償(フリー)クラウドにセンサ値を送り、蓄積し、閲覧し、可視化するなどの共有機能を紹介します。センサネットワークの実現例として参考にしてください。
- ▶ 簡単に、早くクラウドへのデータ送信・蓄積が、実現することができます。
- あらかじめxivelyの登録(メールアドレス、ID、PW設定)およびID関連(「Feed ID」、「Channel」、 それに「API-Key」)の取得を行っておいてください。登録およびID関連の取得は、【添付資料.3】を ご参考ください。
- ここでのサンプルスケッチでは、温度センサ(LM61BIZ)をA0(GND)、A1(Vout)、A2(+Vs)に接続して利用しています。手持ちのセンサなどを使って、スケッチを変更して試してみてください。









17. xively_sample.ino② 【クラウドとの連携】

```
【サンプルスケッチ(cosm sample.ino)】
                                                                                            Feed-ID、CHANNEL、API-Keys
// a3gim sample -- httpPOST (Use xively.com Cloud Service)
                                                                                            は、予めcosm.comに登録し、取得
#include <SoftwareSerial.h>
                                                                                            しておく【添付資料.3】
#include "a3gim.h"
#define LM61BIZ_Pin 1 // LM61BIZ output pin: A1
                                                              Channelが入る
#define baudrate 9600UL
                                        Feed- IDが入る
const char *server = "api.xively.com";
const char *path = "v2/feeds/FEED-ID/datastreams/CHANNEL? method=put";
const char *header="X-ApiKey: API-Keys r$nContent-Type: text/csv$r$n";
int port = a3gimDEFAULT PORT;
                                                                                                           int getTemp(void)
const boolean useHTTPS = true;
                                     Keyが入る
                                                                                                           int mV = analogRead(LM61BIZ Pin) * 4.88;
char res[50]:
                                                                                                            return (mV- 600);
char body[20];
int len;
                                     センサ設定
void setup()
                                                                          void loop()
 pinMode(A0, OUTPUT); // A0(LM61BIZ - GND)
 digitalWrite(A0, LOW);
                                                                           static int Count = 1;
 pinMode(A2, OUTPUT); // A2(LM61BIZ - VSS+)
 digitalWrite(A2, HIGH);
                                                                           Serial.print(">httpPOST requesting: ");
                                                                           Serial.println(Count++, DEC);
 Serial.begin(baudrate);
                                                                           len = sizeof(res);
 delay(3000); // Wait for Start Serial Monitor
                                                                           int temp = getTemp();
 Serial.println(">Ready.");
                                                                           sprintf(body, "%d.%d", temp/10,temp%10);
                                                                           if (a3gim.httpPOST(server, port, path, header, body, res, &len, true) == 0) {
 Serial.print("Initializing..");
                                                                            Serial.println("Succeeded.");
 if (a3gs.start(powerPin) == 0 \&\& a3gs.begin(0, baudrate) == 0)
                                                                            Serial.print(">Response=[");
                                                                                                                httpPOST利用
  Serial.println("Succeeded.");
                                                                            Serial.print(res);
 else {
                                                                            Serial.println("]");
  Serial.println("Failed.");
  while (1); // STOP
                                                                           else
                                                                            Serial.println("Failed.");
                                                                           delay(30000); // take an interval
```

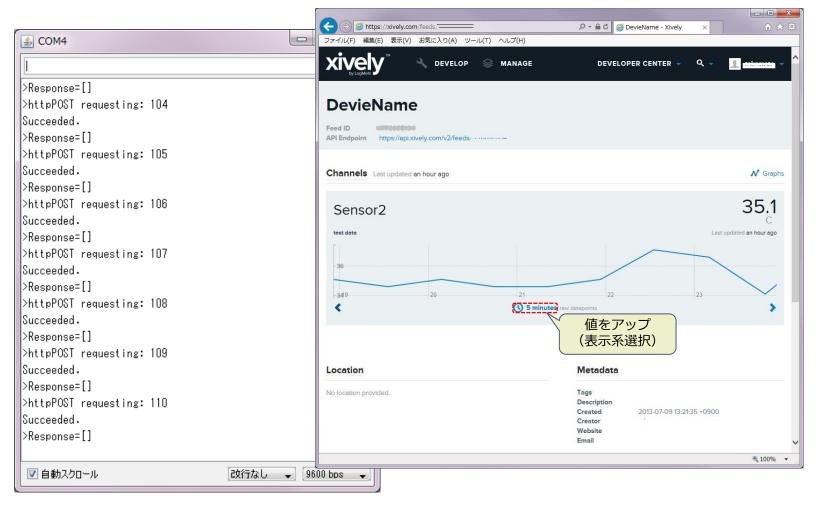






17. xively_sample.ino③ 【クラウドとの連携】

▶ xively_sampleの実行結果事例





【添付資料.1】a3gim.hライブラリー覧表	P.88
【添付資料.2】tweetトークン取得	P.89
【添付資料.3】xivelyの利用登録	P.90
【添付資料.4】語彙説明	P.93
【添付資料.5】電源供給について	P.94
【添付資料.6】サンプル1:メール送信	P.95
【添付資料.7】サンプル2: xively.com利用	P.96

添付資料





【添付資料.1】a3gim.h ライブラリー覧表

※ Arduino GSM/GPRSシールド用ライブラリと互換性がある関数

* 無償サービス「http://arduino-tweet.appspot.com/」を利用(要Twitterの登録)

分類	メソッド名	機能概要	補足
コントロール関係	getStatus**	3Gシールドの状態取得	
	begin [*]	ライブラリの初期化	
	end [*]	ライブラリの終了	
	restart*	3Gシールドのリセット	
	start*	3Gシールドの電源ON	
	shutdown*	3Gシールドの電源OFF	
	getIMEI	IMEIの取得	携带端末固有番号
	setLED	緑色LEDのON/OFF	
	setBaudrate	通信速度の設定	
	setAirplaneMode	エアプレーン(機内)モードのON/OFF	
	setResult	通信結果を取得	
ショートメッセージ関係	sendSMS*	SMSの送信	
	availableSMS**	SMSの受信状態チェック	
(SMS)	readSMS**	SMSの読出し	
	onSMSReceived	SMS着信時のコールバック設定	
インターネット関係 (Web)	httpGET*	GETメソッドの要求	https取得も可能
	httpPOST	POSTメソッドの要求	
	tweet*	Twitterへの投稿	*
インターネット関係 (TCP)	connectTCP*	TCPコネクションを接続する	
	disconnectTCP**	TCPコネクションを切断する	
	writeBegin	シリアル通信で直接書込み	
	read*	データを読み込む	
	write*	データを書き出す	
位置情報取得(GPS)関係	getLocation	現在位置の取得	内蔵GPSを使用
	getLocation2	Assisted GPSを使った現在位置の取得	
その他ライブラリ	getServices	利用可能サービスの取得	
	getRSSI	電波強度の取得	
	getTime	現在時刻の取得	日付・時刻形式
	getTime2	現在時刻の取得	通算秒形式
	getVersion	3Gシールド(gw3gアプリ)のバージョンの取得	
プロファイル関係	setDefaultProfile	デフォルトプロファイル設定	
	getDefaultProfile	デフォルトプロファイル取得	
ATコマンドパススルー	enterAT	ATコマンドパススルーモード	



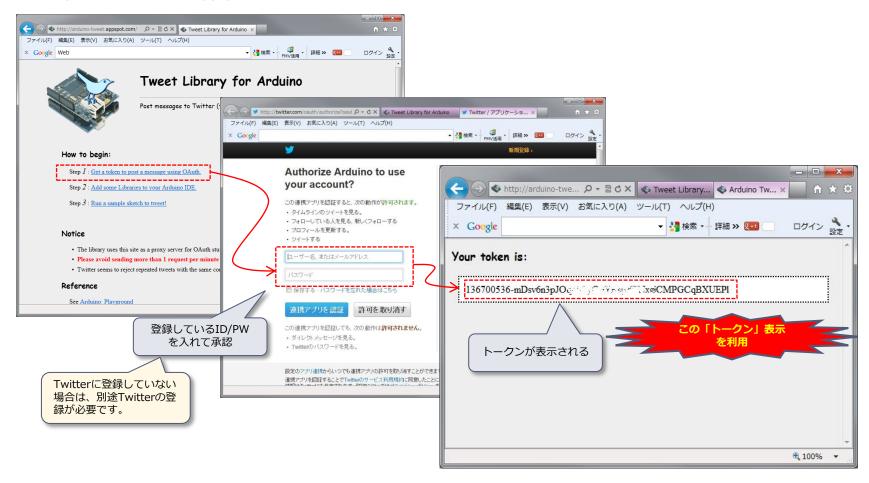




【添付資料.2】tweetトークン取得

▶ 以下のサイトからトークン取得を行う。

http://arduino-tweet.appspot.com/





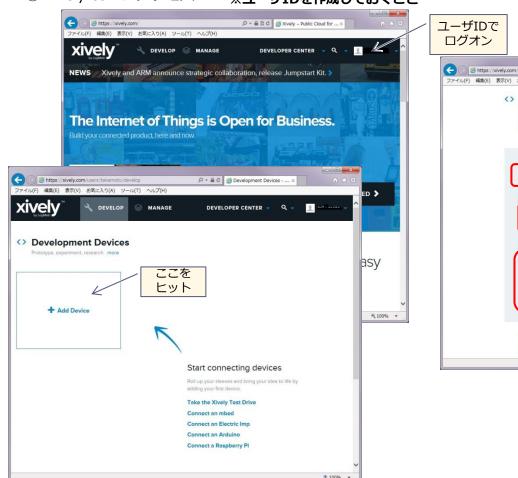




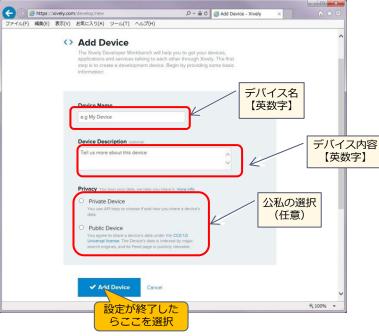
①のメールアドレスに この様なメールが届く

【添付資料.3】 xivelyの利用登録①

① xively.comにアクセス ※ユーザIDを作成しておくこと



③ 「Add Device」の詳細設定



② 新たにデバイス設定「+Add Device」を選択

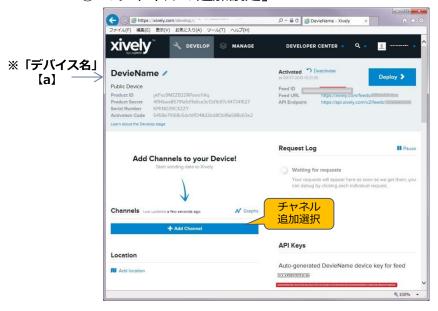




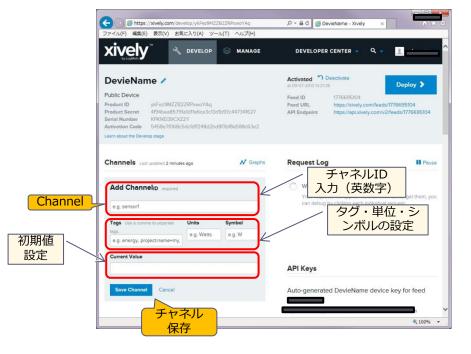


【添付資料.3 】 xivelyの利用登録②

④ 「チャネルの追加設定 |



⑤チャネルの詳細設定



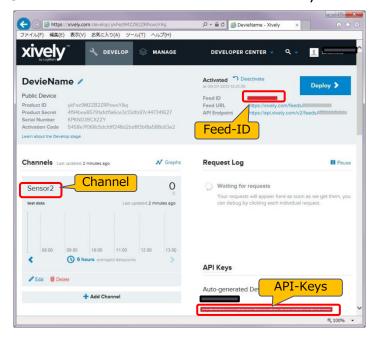






【添付資料.3 】 xivelyの利用登録③

⑥ 設定されたFeed-IDやChanel、API-Keys の確認



※ Feed-IDとAPI-Keysはシステム側で設定。 Channelは、ユーザが設定。







【添付資料.4】語彙説明

- シールドとは、Arduino本体に装着できる拡張ボードのことを指す。
- スケッチとは、Arduino上で稼働するプログラムのこと。
- ▶ レシピとは、Arduino上での配線図やスケッチのこと。作りながら考える意味で、「秘訣」などの意味も含まれる。
- ▶ プロファイルとは、SIMカードを認識するためのもの(APN)で、予めIEMモジュールに、どのSIMカードで利用するかを設定する必要がある。(省略時設定は、別途記載している通り)
- a3gimとは、arduino 3GIM の略で、Arduino上で3gシールドを利用するためのライブラリ群となる。
- ▶ 3GIMとは、3G IoT Module の略で、世界最小の3G通信機器(タブレイン製)となる。
- APN(プロファイル)とは、Access Point Name(アクセス・ポイント・ネーム)の略で、携帯電話ネットワーク上のデータ通信で必要となる接続先を指定する文字列となる。通信キャリアごとやMNVO(仮想移動体通信事業者)ごとに固有の名前が付けられている。
- ▶ Assisted GPSとは、基地局の携帯電話通信網を使ってGPS衛星情報を先に得て、位置情報取得を支援するもの。
- ▶ GPSとは、Global Positioning Systemの略で、高度約2万Kmの複数のGPS衛星から電波を受信し、緯度と経度を割り出す システム。
- GLONASSとは、ロシア政府による衛生測位システムのこと。
- ▶ gw3gとは、GateWay to 3Gの略で、 3 GIM V2.0上のLPC812に組み込んだファームウェア(ライブラリ群)のこと
- ▶ IDEとは、統合的な開発環境のことを指す。
- IMEIとは、International Mobile Subscriber Identityの略で、携帯端末の識別番号で、3 GIM のIMEIは3 G通信モジュールの固有識別番号となる。
- MVNO(仮想移動通信事業者)とは、Mobile Virtual Network Operatorの略で、携帯電話やPHSなどの物理的な移動体回線網を自社で持つことなく、通信キャリアから借りて、自社ブランドで通信サービスを行う事業者のこと。
- RSSI値とは、Received Signal Strength Indicatorの略で、「受信信号強度」のことを意味し、受信機入力に入る受信信号の強度を示す数値となる。
- ▶ UNICODEは、世界的な標準の文字コードのひとつである。
- ▶ UTF-8は、UCSとUNICODEで使える8ビット符号単位の文字符号化形式及び文字符号化スキーム。

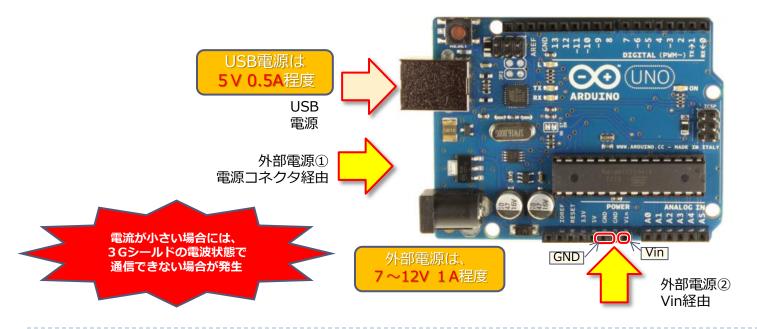






【添付資料.5】電源供給について

- ▶ 3Gシールドは、これまでの調査で、一時的に約200mA~600mAの電力を消費することが分かりました。
- ▶ このことで、電波状態やセンサ類の利用などの状況によっては、PCによるUSB電源(5 V500mA)では、電源供給できない状態になり、通信できない場合が発生します。この障害の場合には、外部電源(推奨は9 V1.3A)を別途、取って頂くことで、通信が可能となります。
- ▶ この他、USB接続ではなく、外部電源のみで利用される場合には、消費電力(特に電流)をチェック(測定)して、安定した バッテリー電力を使ってご使用ください。(購入時に提供される別資料「3Gシールド&3GIM利用のための電源供給につい て」を参照ください)
- ▶ 二次電池(ニッケル水素やリチウムイオン)などのバッテリーをご利用される場合には、電池の電圧・電流(その積による電力)能力を十分満たすものを外部電源としてご利用ください。外部電源は、電源コネクタ部分、あるいはVinとGNDの2つのピンから供給することができます。







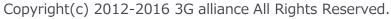




【添付資料.6】サンプル1:メール送信



http://www.kumikomi.net/archives/2013/08/ca053gcc.php



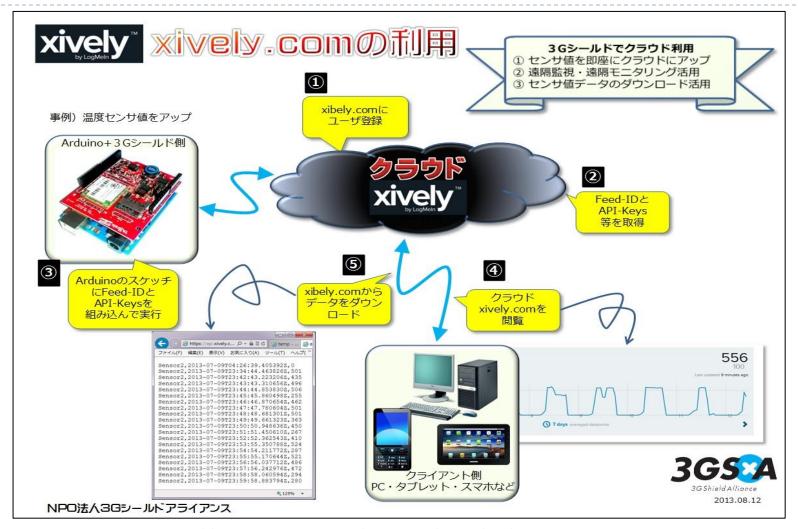








【添付資料.7】サンプル2:xively.com利用



http://www.kumikomi.net/archives/2013/08/ca063gcc.php



