

# IoTプラットフォーム活用勉強会#2

【BluemixにArduino からデータを送ろう】

平成28年3月25日

ソフトピアジャパン ドリーム・コア1F ネクストコア

# MQTTプロトコル

MQTT (Message Queueing Telemetry Transport) は、IoTのために設計された軽量で高速の通信プロトコルです。

ArduinoとBluemixを接続する際に利用できます。

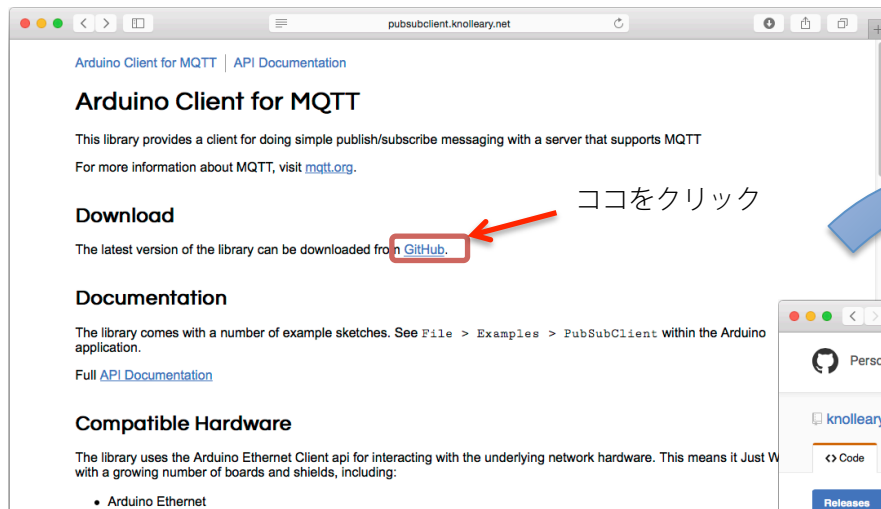


MQTT V3.1 プロトコル仕様

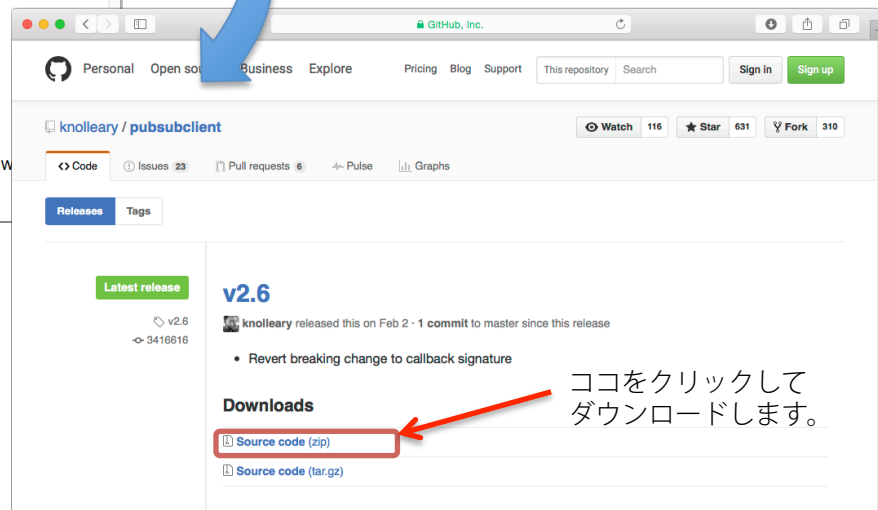
[http://public.dhe.ibm.com/software/dw/jp/websphere/wmq/mqtt31\\_spec/mqtt-v3r1\\_ja.pdf](http://public.dhe.ibm.com/software/dw/jp/websphere/wmq/mqtt31_spec/mqtt-v3r1_ja.pdf)

# Arduino MQTT ライブラリー

ArduinoでMQTTを使用するためのライブラリをダウンロードします。



ココをクリック



ココをクリックして  
ダウンロードします。

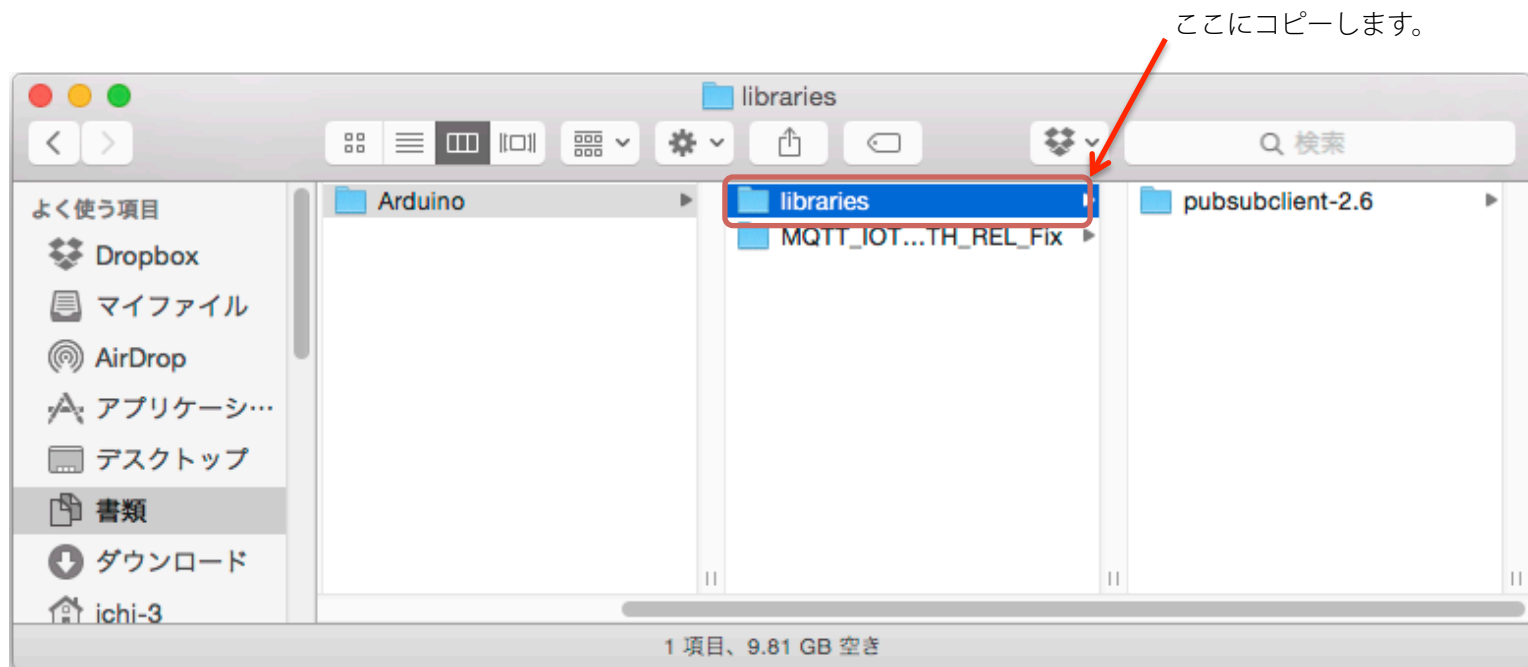
<http://pubsubclient.knolleary.net>

<https://github.com/knolleary/pubsubclient/releases/tag/v2.6>

# ライブラリの導入

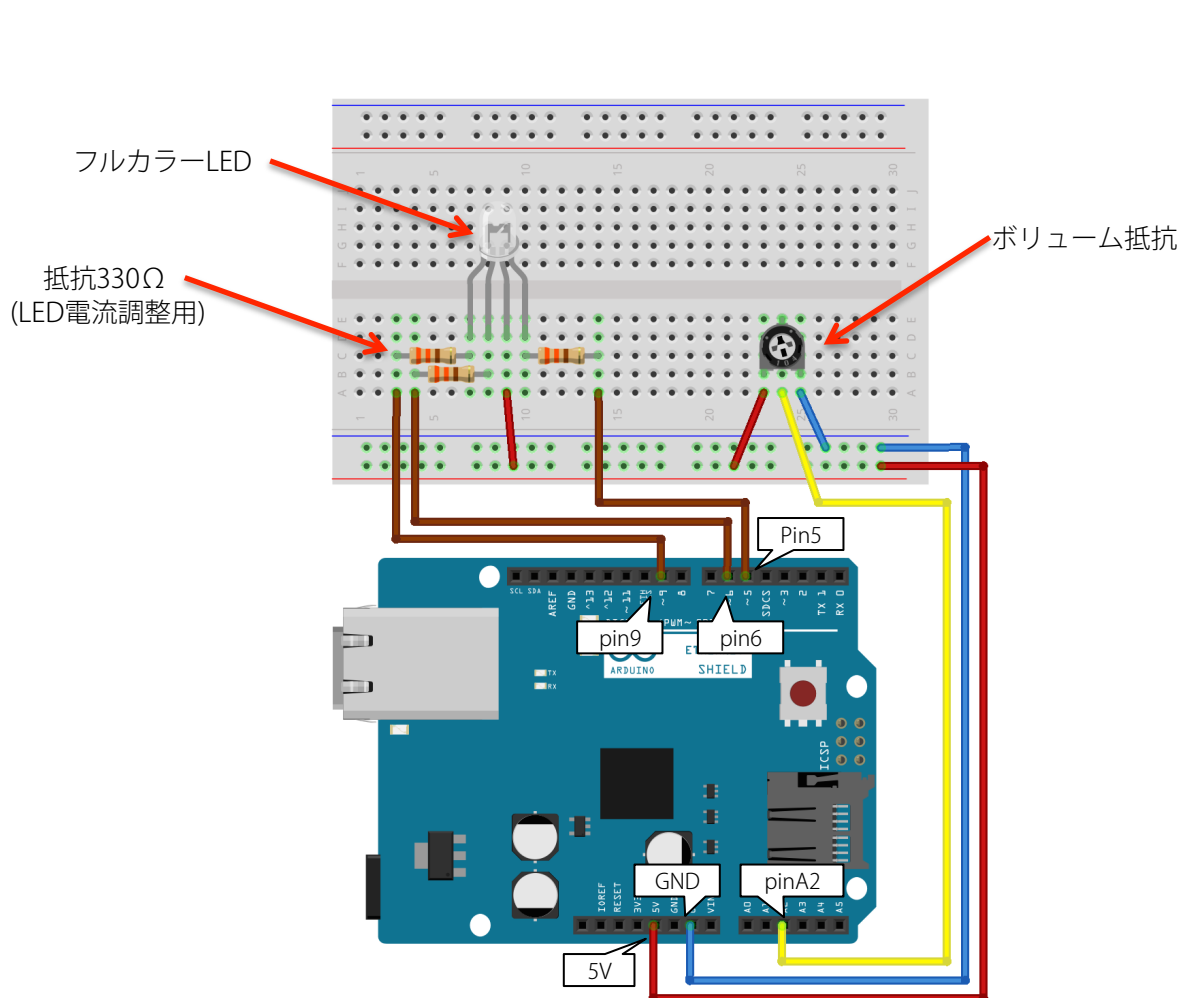
ダウンロードしたライブラリ「pubsubclient-2.6」は  
「Arduinoフォルダ」直下の「libraries」フォルダにコピーします。

コピー後にArduinoIDEを再起動するとメニューの  
「スケッチ」→「include Library」  
に「pubsubclient」が追加されます。

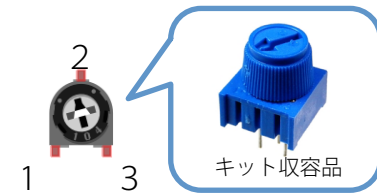


# Bluemix-Arduino連携：回路

入力用センサとしてボリューム抵抗、状態表示用にフルカラーLEDを使用します。

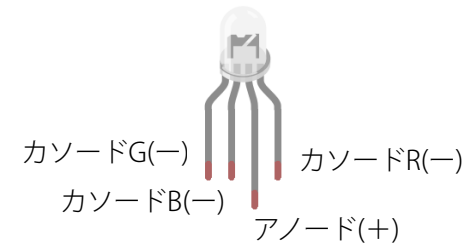


ボリューム抵抗10KΩ



ピン1を電源(5V)、  
ピン2をアナログ入力ピン、  
ピン3をGNDへ繋がます。

フルカラーLED(アノードコモン)



抵抗330Ω



# Bluemix-Arduino連携：コード①

サンプル「Bluemix\_Arduino.ino」を読み込み以下の下線の箇所を書き換えます。

```
#include <SPI.h>
#include <Ethernet.h>
#include <PubSubClient.h>
```

Arduino イーサネットシールド2を使用の場合は  
#include <Ethernet2.h>と記述する。  
(IDE1.6.\*系列の場合はIDE1.7\*系列よりライブラリを移植する必要あり)

```
// Update this to either the MAC address found on the sticker on your ethernet shield (newer shields)
// or a different random hexadecimal value (change at least the last four bytes)
```

```
byte mac[] = {0x90, 0xA2, 0xDA, 0x0F, 0xFB, 0x21};
char macstr[] = "90a2da0ffb21";
```

MACアドレスを指定

MACアドレスを連結した文字列(小文字)

```
char servername[]="w15sdv.messaging.internetofthings.ibmcloud.com";
String clientName = String("d:w15sdv:Arduino:") + macstr;
String topicName = String("iot-2/evt/status/fmt/json");
char username[]="use-token-auth";
char password[]="9wVx0Lj5+BYer*rGas";
```

Bluemixでの設定を反映させる。

※赤字の下線の箇所を編集

```
EthernetClient ethClient;
```

```
const int analogInPin = A2; // アナログ入力ピン
int sensorValue01 = 0; // センサ読み取り値
const float s = 1.0; // 彩度
const float v = 0.3; // 明度
int r,g,b; // RGB値
const int PinR = 5;
const int PinG = 9;
const int PinB = 6;
```

フルカラーLEDの  
明度・彩度は固定

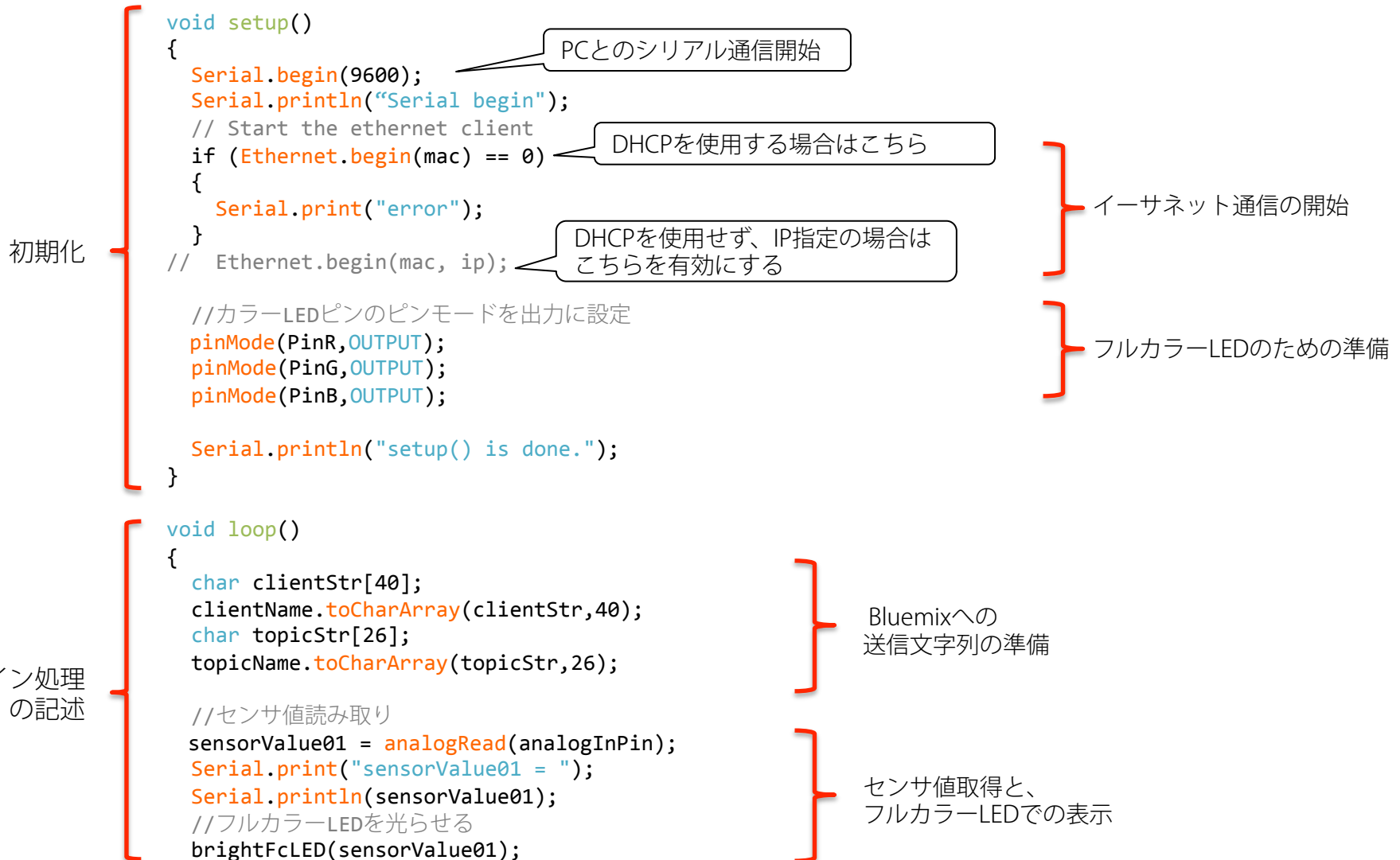
フルカラーLED  
制御ピン番号

センサおよび  
フルカラーLEDの設定

```
// MQTTクライアントのインスタンスの宣言&作成
PubSubClient client(servername, 1883, 0, ethClient);
```

各種  
定義・宣言

# Bluemix-Arduino連携：コード②



# Bluemix-Arduino連携：コード③

```
if (!client.connected()) {  
  Serial.print("Trying to connect to: ");  
  Serial.println(clientStr);  
  client.connect(clientStr,username,password);  
}  
if (client.connected() ) {  
  String json = buildJson();  
  char jsonStr[200];  
  json.toCharArray(jsonStr,200);  
  boolean pubresult = client.publish(topicStr,jsonStr);  
  Serial.print("attempt to send ");  
  Serial.println(jsonStr);  
  Serial.print ("to ");  
  Serial.println(topicStr);  
  if (pubresult)  
    Serial.println("successfully sent");  
  else  
    Serial.println("unsuccessfully sent");  
}  
delay(3000);  
}
```

Bluemixに接続されていない  
場合は再接続

Json文字列の作成

StringオブジェクトをChar型の配列に変換

送信

送信結果をシリアルへ表示

Bluemixに接続出来た  
場合の処理

メイン処理  
の記述

つづく



# Bluemix-Arduino連携 : コード④

ユーザ定義  
関数①

```

/*//////////////////////////////////////////////////////////////
Bluemixへの送信用Json文字列を作成する関数
//////////////////////////////////////////////////////////////*/
String buildJson() {
  String data = "{}";
  data+="\n";
  data+= "\"d\": {";
  data+="\n";
  data+= "\"myName\": \"Arduino Sensors\",";
  data+="\n";
  data+= "\"Sensor01\": ";
  data+=(int)sensorValue01;
  data+="\n";
  data+= "}";
  data+="\n";
  data+= "}";
  return data;
}

```

ここで作っている文字列の例：

```

{
  "d": {
    "myName": "Arduino Sensors",
    "Sensor01": 985
  }
}

```

センサの値(sensorValue01)

ユーザ定義  
関数②

```

/*//////////////////////////////////////////////////////////////
引数を元にフルカラーLEDを(赤～青に)光らせる関数
//////////////////////////////////////////////////////////////*/
void brightFcLED(int sensorVal){
  //センサの値を色相の赤～青(H:0~239)に変換する
  int h = map( sensorVal , 0 , 1023 , 0 , 239);

  //HSV(色相・彩度・明度)からRGBを計算
  hsv2rgb( h , s , v);

  analogWrite( PinR , 255 - r);
  analogWrite( PinG , 255 - g);
  analogWrite( PinB , 255 - b);
}

```

HSV形式の色指定からRGB値をを計算する関数

アノードコモンのフルカラーLEDのため、  
255からの差を指定する。  
(トータル255で消灯、0でフル発光)

# Bluemix-Arduino連携 : コード⑤

```

/*/////////////////////////////////////////////////////////////////
HSV(色相・彩度・明度)からRGBを計算する関数
RGBの戻り値はグローバル変数に記述
////////////////////////////////////////////////////////////////*/
void hsv2rgb(int h , float s , float v){
    int i , lr , lg , lb , vi;
    int p1 , p2 , p3;
    float f;
    i = (int)(h / 60.0);
    f = h / 60.0 - i;
    p1 = (int)(v * (1.0 - s) * 255.0);
    p2 = (int)(v * (1.0 - s * f) * 255.0);
    p3 = (int)(v * (1.0 - s * (1.0 - f)) * 255.0);
    vi = (int)(v * 255.0);
    if(i == 0) {lr = vi ; lg = p3 ; lb = p1;}
    if(i == 1) {lr = p2 ; lg = vi ; lb = p1;}
    if(i == 2) {lr = p1 ; lg = vi ; lb = p3;}
    if(i == 3) {lr = p1 ; lg = p2 ; lb = vi;}
    if(i == 4) {lr = p3 ; lg = p1 ; lb = vi;}
    if(i == 5) {lr = vi ; lg = p1 ; lb = p2;}
    r = lr;
    g = lg;
    b = lb;
}

```

ユーザ定義  
関数③

HSVからRGBへの変換は以下のサイトを参考にしました。

- [http://maicommon.ciao.jp/ss/Arduino\\_g/PWM/index.htm](http://maicommon.ciao.jp/ss/Arduino_g/PWM/index.htm)
- <http://www.peko-step.com/tool/hsvrgb.html>

# 補足：参考URL

- Arduino Uno と IBM IoT Foundation を利用してクラウド対応の温度センサーを作成する  
<http://www.ibm.com/developerworks/jp/cloud/library/cl-bluemix-arduino-iot1/index.html>

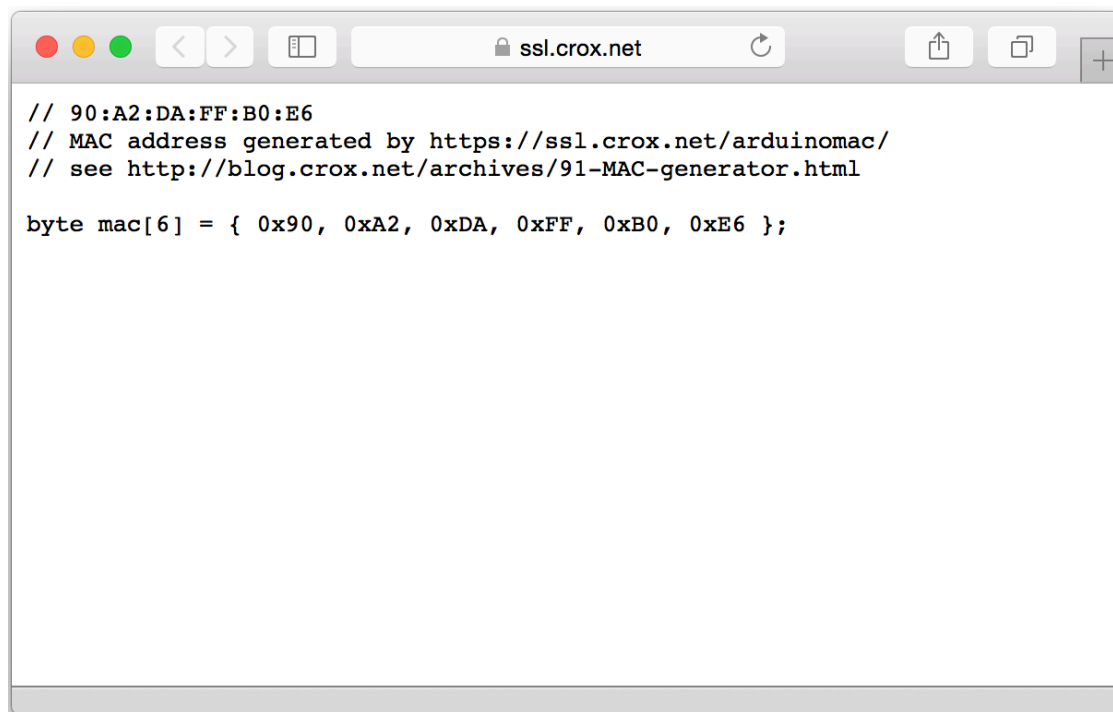
- ArduinoをBluemixに接続するチュートリアルTips  
<http://qiita.com/sotoiwa/items/5f9f475a8051ed9f0f52>

- MQTT V3.1 プロトコル仕様  
[http://public.dhe.ibm.com/software/dw/jp/websphere/wmq/mqtt31\\_spec/mqtt-v3r1\\_ja.pdf](http://public.dhe.ibm.com/software/dw/jp/websphere/wmq/mqtt31_spec/mqtt-v3r1_ja.pdf)

# 補足：互換品のMACアドレス

イーサネットシールドの互換品にはMACアドレスが添付されていません。ローカルネットワーク上でユニークなMACアドレスをソースコード上で設定すれば良いため、利用可能なMACアドレスを生成して利用します。

MACアドレス生成サイト「<https://ssl.crox.net/arduinomac/>」

A screenshot of a web browser window. The address bar shows 'ssl.crox.net'. The page content displays a generated MAC address and the corresponding C code for an Arduino. The text is as follows:

```
// 90:A2:DA:FF:B0:E6
// MAC address generated by https://ssl.crox.net/arduinomac/
// see http://blog.crox.net/archives/91-MAC-generator.html

byte mac[6] = { 0x90, 0xA2, 0xDA, 0xFF, 0xB0, 0xE6 };
```

# 補足：イーサネットシールド2の場合

イーサネットシールド2はIDEのver1.7.\*系列でしかサポートされていません。イーサネットシールド2を使用する場合はver1.7.\*系列のインストールもしくはver1.7.\*系列のライブラリをver1.6.\*系列のライブラリに移植する必要があります。

IDEのver1.7.\*系列のダウンロード

<http://www.arduino.org>

参考：Arduino戦争：グループ分裂、そして新製品の登場

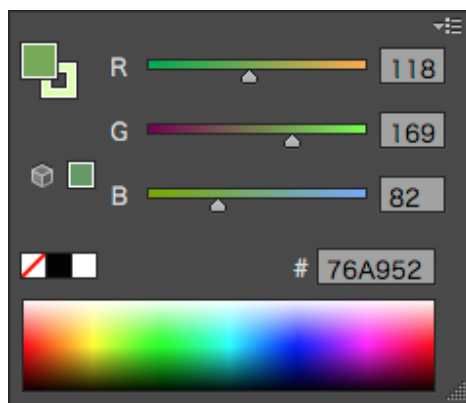
<http://makezine.jp/blog/2015/03/arduino-vs-arduino.html>

# 補足：RGBとHSVによる色の指定

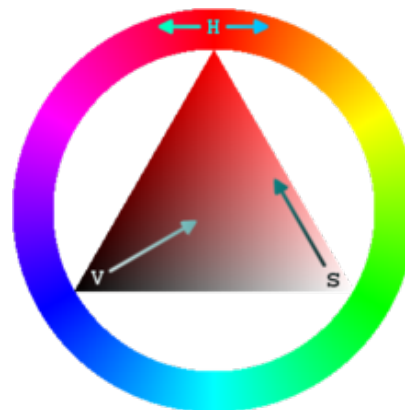
フルカラーLEDは赤(R)緑(G)青(B)三色のLEDから構成されており、その色を決めるには、三色それぞれの発光具合を指定して行います。これをRGB方式といいます。ハードウェア的な仕組みから言えば合理的な方法ですが、人の感覚ではイメージし難い方法です。

一方、色相(H)彩度(S)明度(V)で色の定義を行うHSV方式は人がイメージしやすく、連続して色の変更を行う場合などに扱いやすいものです。

そのため、本サンプルではHSVからRGBへ変換する関数を使用しています。



RGBによる色指定



HSVによる色指定

下記サイトのHSV→RGB変換を参考にしています。

[http://maicommon.ciao.jp/ss/Arduino\\_g/PWM/index.htm](http://maicommon.ciao.jp/ss/Arduino_g/PWM/index.htm)

<http://www.peko-step.com/tool/hsvrgb.html>

# 補足：電子部品通販

よく利用される電子部品の通販サイトです。

- ・スイッチサイエンス

<https://www.switch-science.com>



Arduinoといえばここ。各種センサ、サーボも豊富、価格高め。  
Raspberry Piも買えます。

- ・RSコンポーネンツ

<http://jp.rs-online.com/web/>



Raspberry Piの本家

- ・秋月電子通商

<http://akizukidenshi.com/>

**秋月電子通商**

電子部品一般。

ACアダプタ、ユニバーサル基板、LED、振動モータなどが豊富  
オリジナルのキットが豊富(安定化電源キットなど)

# 補足：電子部品通販

- ・ ストロベリー・リナックス

**Strawberry Linux**

<http://strawberry-linux.com>

電源関係(DC-DCコンバータ)、  
ステッピングモータドライバなどのオリジナル基板が得意

- ・ 千石電商

 **せんごくネット通販**

<http://www.sengoku.co.jp>

電子部品一般。配線材、コネクタ類が秋月より豊富。

- ・ オヤイデ電気

 **オヤイデ電気**  
**oyaide.com**

<http://www.oyaide.com/ja>

ケーブル専門店

- ・ HobbyKing

 **HobbyKing**  
**.com**

<http://www.hobbyking.com/hobbyking/store/index.asp>

ラジコンパーツ店。サーボが豊富で安い。  
ただし、海外のため送料・関税に注意。