

Arduinoで センサ・データをクラウドへ

2014年6月24日

先端|T活用推進コンソーシアム クラウド・テクノロジー活用部会 荒本 道隆

本日の注意事項



- ・<u>ハードウエアを扱うため、絶対厳守</u>
 - -回路を変更する時は、電源(USB)を抜く
 - 個人所有の部品は、自己管理してください
 - ・
 能に貸したか、どこにあるか
 - 「壊したかも?」と思ったら、正直に言おう
- 台数が予想以上に足りませんでした
 - LED, 照度センサ、イーサネット/無線
 - 複数人でデバイスの共有をお願いします
 - ・書き込み時だけ、自分のPCにArduinoを刺す
- 経験者は、フォローに回ってください
 各班に、経験者が入るようお願いします

Arduino とは



Arduino はスタンドアロン型のインタラクティブデバイス開発だけでなく、ホスト コンピュータ上のソフトウェア(例えば、Adobe Flash、Processing、Max/MSP. Pure Data、SuperCollider)で制御することもできる。オープンソースハード ウェアでありハードウェア設計情報のEAGLEファイルは無料で公開されてお り、組み立て済みの基板を購入することもできるほか、誰でも自分の手で Arduino を組み立てることができる。 Arduinoプロジェクトは2005年にイタリアで始まり、当時入手可能だった他の 学生向けのロボット製造用コントロールデバイスよりも安価なプロトタイピン グ・システムを製造することを目的にスタートした。設計グループは多くの競 合製品よりも遥かに安価で簡単に使用できるプラットフォームの開発に成功し た。Arduinoボードは、2008年10月までに5万ユニット以上³³が、2011年2月 で約15万台^国販売されている。Arduinoプロジェクトは2006年度のアルス・エ レクトロニカ賞で名誉言及を受けている。[5][6][7]



ウィキペデアより

Arduinoの特徴



- アナログ・デジタルの入出力が複数ある
 そこにセンサやリレーを簡単に接続できる
- 豊富なシールド
 - イーサネット, GPS, LCD, モーター制御
 - <u>http://ideahack.me/article/147</u>
- センサ以外は、使い慣れた技術・用語が多い
 USB, シリアル, HTTP, TCP/IP
- 取り扱いが容易
 - 不器用な私でも何とかなった
 - ・ 無線シールドの半田付けに失敗し、3つほど捨てたけど
 - そんなに高くない
 - ・ 壊してしまっても、大人なら平気

機器購入時の注意事項



- 無線LAN, Bluetoothは、技適が付いているか?
 - 海外の無線シールドには、技適が付いていない
 - 技適が付いていないものを使うデメリット
 - 発表時に、構成を詳しく言えない
 - **電波法違反**=「1年以下の懲役又は100万円以下の罰金刑に処せれる」
 - 2020年の東京オリンピックで、海外旅行者のスマホをWiFi接続するために、何らかの法改正があるかも?
 - <u>http://internet.watch.impress.co.jp/docs/news/20140613_653369.html</u>
 - 最初は有線→無線変換が簡単
 - 配置
 - 電源の確保
 - センサだけなら、電池でも結構持つ
 - ・ 無線LANを電池で使いたいのなら、省電力のものを選択

– パッケージング

- ・ タッパーに穴を開けて、突っ込んじゃうのが簡単
- 大きなセンサは固定しておかないとグラつく

Arduinoの概要



デジタル入出力(プログラムで切り替え)

アナログ入力(0~1023の範囲)



電源

ブレッドボードの概要





プロトタイピング





ドライバと開発環境をインストール



- ・ ダウンロード
 - http://arduino.cc/en/main/software
- Windows
 - 「Windows ZIP file」をダウンロードして、解凍
 - drivers¥arduino.infを右クリックして「インストール」
 - Arduino を接続
 - arduino.exe でIDEを起動
 - メニューの「ツール」から
 - ・ →「シリアルポート」→「COM3」(PCによって違う)を選択
 - →「マイコンボード」→「Arduino Uno」を選択
- Mac
 - 「Mac OS X」をダウンロードして、解凍
 - Arduino を接続
 - arduino でIDEを起動
 - メニューの「ツール」から
 - ・ →「シリアルポート」→「/dev/tty.usbmodem3d11」を選択
 - →「マイコンボード」→「Arduino Uno」を選択





LEDを点ける

 - 状況が目視できるようになる



ステップ1-1



- LEDを1秒ごとに点滅させる

 まずは、デジタルの8番をLEDの+に接続
 - さっきまでの5Vの線は外す





ステップ2-1



- ・ 照度センサの値をPCで参照
 照度センサをアナログの0番に入力する
 - 足の短い方:5V
 - ・ 足の長い方:GND→抵抗→、A0に渡す



ステップ2-2



照度センサの値をPCで参照 - プログラムを作成 - 実行後は、「ツール」→「シリアルモニタ」で確認 #define LED OUTPUT 8 #define CDS INPUT 0 boolean led = true; void setup() { Serial.begin(9600); void loop() { int val = analogRead(CDS INPUT); Serial.print ("CdS :"); Serial.print (val); Serial.println(); delay(1000);

ステップ3



「暗くなったら、LEDを灯す」を実現
 – ステップ2のプログラムを改良



その他のセンサ



- ・人感センサ
 - Panasonic AMN41121
- 温湿度センサ
 - DHT11



センサデータをクラウドに投げる-1



- 送信先URL
 - 今回の勉強会専用
 - ・しばらくすると、削除します
 - <u>http://219.101.208.155/arduino/</u>ID/....
 - IDに自分の識別子を付ける
 - サーバ側は何も処理しないので、パラメータを適当に足す
 - あまり大量に投げると、他の人が見にくくなるので注意
- 確認用URL
 - <u>http://219.101.208.155/arduino/access_log.view.cgi</u>
 - ・最新の100件の送信ログを参照
 - 必要に応じて、「ID=xxxx」専用を作ります





- イーサネットシールドでHTTP通信
 Macアドレスをハードコード、LAN内で重複しないように
 - DNSを参照するのは手間なので、直IPアドレスで指定
 - 別機器で無線LAN化(コンバータモードを使用)



イーサネットシールド



無線LAN化機器 PLANEX MZK-PR150N





- プログラムをちょっと改良
 - 頻繁に値の書き換えをしないようにする
 - 通信量を減らすため

```
void loop() {
#define LED OUTPUT 8
                                            int val = analogRead(CDS INPUT);
#define CDS INPUT 0
                                            boolean writeFlag = false;
                                            if (val < 400){ // 暗ければ
boolean led = LOW;
                                             if (led != HIGH){
                                              led = HIGH: // 点ける
void setup() {
 pinMode(LED_OUTPUT, OUTPUT);
                                              writeFlag = true;
// Serial.begin(9600);
                                            } else { // そうでなければ、
                                             if (led != LOW){
                                              led = LOW; // 消す
                                              writeFlag = true;
```

```
writeFlag = true;

}

if (writeFlag == true){

digitalWrite(LED_OUTPUT, led);

}

delay(100); // 反応の遅延を減らす
```

センサデータをクラウドに投げる-4



プログラムをHTTP通信に対応(前半)

```
#include <SPI.h>
#include <Ethernet.h>
#define LED OUTPUT 8
#define CDS INPUT 0
boolean Ied = LOW;
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // 重複しないように
IPAddress server(219,101,208,155);
int port = 80;
void setup() {
 pinMode(LED OUTPUT, OUTPUT);
 Serial.begin(9600);
 Serial.println("DHCP Request");
 for(;Ethernet.begin(mac) == 0;) { // DHCPサーバからIPアドレスを取得
  Serial.println("Failed to configure Ethernet using DHCP");
  delay(10000);
 Serial.println("START!");
```

センサデータをクラウドに投げる-4



プログラムをHTTP通信に対応(後半)

```
void loop(){
 int val = analogRead(CDS_INPUT);
 boolean writeFlag = false;
 if (val < 400){ // 暗ければ
  if (led != HIGH){
                                          EthernetClient client;
   led = HIGH; // 点ける
                                          if (client.connect(server, port)) {
   writeFlag = true;
                                           Serial.println("HTTP connected");
                                           // Make a HTTP request:
          // そうでなければ、
 } else {
                                           client.print("GET /arduino/ID/");// IDを識別できる名前に変更
  if (led != LOW){
                                           client.print("?LED=");
   led = LOW; // 消す
                                           client.print(led);
   writeFlag = true;
                                           client.print("&cds=");
                                           client.print(val);
                                           client.print(" HTTP/1.0");
 if (writeFlag == true){
                                           client.println();
  digitalWrite(LED_OUTPUT, led);
                                           client.println();
  Serial.print ("CdS :");
  Serial.print (val);
                                           delay(100); // 送信が終わるまで、ちょっと待つ
  Serial.println();
                                           client.stop();
                                        delay(100);
```

時間が余ったら



- テスト配置用の配線にする
 -『ブレッドボードの概要』ページ参照
- ・ サーバ側の値によって、LEDを点灯/消灯
 - 1秒間隔でGETを行う
 - レスポンスの内容でLEDを制御
 - 本番運用時はCometを利用
- 別のセンサも使ってみる
 - 人感センサ
 - ネットで面白そうなものを検索
 - センサ以外
 - デジタル、アナログ
- ・他に必要なもの
 - セキュリティ(更新時、参照時)

後片付け



- 持ってきた物は、全部そろっていますか?
- センサやショートピンが落ちていませんか?
- ・最初の状態に戻す

