近畿大学短大論集
第46巻 第1号（2013年12月）
p.61～69

可搬型環境測定システムの開発

黒　田　正治郎

抄録

温度、湿度、気圧、光量、放射線量、磁気などの環境量を測定する環境測定システムの試作を行った。温度センサや湿度センサ、GPSモジュールなどとマイクロコンピュータとスマートフォン、パソコンを組み合わせることにより、小型軽量、低コスト、拡張性に優れたシステムとした。システムの最適化により、13時間の連続計測と4か所同時計測がリアルタイムで可能になった。

キーワード

可搬性、環境測定システム、スマートフォン、プログラミング、Arduino

The Development of the Environmental Data Measurement System with High Portability

Kuroda, Shoziro

Abstract

Using a micro-computer, a smart phone and several sensors, I experimentally devised a system which is able to measure the environmental data such as temperature, humidity and atmospheric pressure, etc.

This system is characterized by high portability, high-cost-performance and high versatility. By optimizing the system, it is possible to measure 3 environmental data in real time consecutively for 13 hours at four different places.

Key Words

high portability, environmental data measurement system, micro-computer, smartphone, programming, Arduino

目　　次

近畿大学短期大学部教授
2013年9月28日受理

§1．はじめに

物理計測において、温度、湿度、気圧、光量、放射線量、磁気などの実験環境における環境量を測定することは重要な要素であり、リアルタイムでの集計を必要とする場合が多い。そのために、専用の測定機器の準備と設置が必要になり、多くの時間を費やす事がある。単体で高性能な測定器や複合型の機器は多く開発されているが、いずれも、実験に必要な環境量を総合的に計測できる機器は少ない。また、大気汚染物質であるPM2.5や黄砂などを監視する大気汚染物質広域監視システム：そらまめくん[1]や、花粉の飛散状況や予測を行う環境省花粉観測システム：はなこさん[2]といった広域環境を計測するシステムもあるが、いずれも独自に開発したシステムに組み込むことは難しい。

そこで、複数のセンサとパソコン(以降PC)、スマートフォン(以降SP)を組み合わせて、安価で可搬性に優れた環境測定システムを試作した。本システムは、温度、湿度、気圧などの環境量を総合的に計測するもので、センサの組み合わせにより使用目的を容易に変更でき、GPSから位置情報を取得することにより、実験場所や時間を特定することができるシステムとした。

§2．システム設計

環境測定システムは設置する場所により、次の2つの仕様が考えられる。①環境量を測定する場所の近くに環境計測システムとPCを設置できる場合、②環境計測システムとPCが離れている場合である。①の場合、センサ-PC間をWi-Fiなどの近距離通信やケーブル接続が利用可能であり、②の場合には、長距離通信の環境が不可欠である。そこで、本稿では長距離通信システムを組み込むことにより、①②いずれの場合にも対応できる環境測定システムの開発を目指した。

環境測定システムを設計する前に、現在利用可能な情報処理機器より、PC、SPやTablet、マイクロプロセッサ（以降マイコン：μP）を中核とした場合の特徴の比較を行った。PC単体での開発では、汎用性に優れてはいるものの可搬性と開発費の面で劣る。また、SPのみでは、可搬性は高いものの表示画面が4～7インチと小さく表示情報に制限が生じる。さらに、搭載センサに制約があるために、SP単体でのシステム構築は難しい。同様に、専用機器は、性能に優れ、独自仕様を設計できるが、開発時間と経費面で大きく劣る。いずれの場合も単体のシステムでは、目標とする性能を実現することは難しい。

表1　情報機器の比較

	PC	SP	μP	専用機器
可搬性	△	○	◎	◎
汎用性	◎	△	◎	◎
拡張性	△	△	◎	◎
通信機能	△	◎	△	△
開発費	△	◎	◎	×
開発時間	△	◎	◎	×

そこで、PCの汎用性、SP/Tabletの通信機能、μPの可搬性と拡張性など、各機器の優位な部分を組み合わせてシステム設計を行った。すなわち、μPに環境計測用センサを接続し、環境量の計測と計測したデータのA/D変換を行い、近距離通信機能を利用して計測データをSPに転送する仕様とした。さらに、SPでは受信した計測データの暗号化を行い、メール通信機能によりPCへ計測データを送信した。

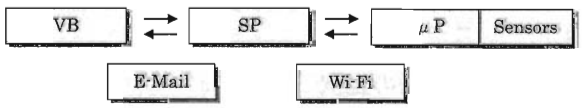

図1　システムの概略

§3．使用機器の選択

1．計測用小型μPの選別

現在、小型のマイコンが多く開発されているので、これを計測用マイコンとして用いる。BASIC Stamp[3]、Propeller[4]など数種類の中から、試作

を繰り返した結果、PC や SP との接続の容易さ、センサやモジュール、シールドの種類の豊富さ、小型、軽量、安価という要素から Arduino[30-33]とした。また、センサは駆動電圧の関係で、一部 SPI 規格のセンサも含むが、基本的には I2C 規格に統一した。

2．SP の選別

計測用小型 μP からの環境データを、PC へ暗号化メールで送信することが主たる目的であるので、簡単なプログラム処理により情報の加工と通信用アプリケーションの開発ができ、近距離通信 Wi-Fi と高速通信網が利用可能であれば、特に機能的な制限はない。実験には、Google 製 Nexus7 と Android 4.2 を搭載した SP を使用した。

3．PC の選別

プログラミング環境と通信環境が整っていれば、通常仕様の PC でシステム構築上の問題はない。本稿では、Windows7＋ Core-i7(3.4GHz)を使用した。

§4．開発言語の選択

1．Arduino でのスケッチ開発

PC 上に Arduino 総合開発環境である Arduino IDE を展開し、C言語を基本に開発された Arduino 言語でスケッチを開発した。完成したスケッチは、通信ポートを使用して USB 接続により Arduino に転送し内蔵 ROM に書き込んだ。

2．SP でのアプリケーション開発

スマートフォン用プログラミング言語として公開されている BASIC![5]、FkmBASIC[6]、Mobile BASIC[7]、JavaScript[34-35]、Perl[36] などによる試作の結果、Bluetooth 接続の安定性、E-mail 機能を有すること、サーバの使用制約の少なさ、PC との連携性の高さから Python[8] とした。SP へは、多言語との混在が可能で統一された開発環境を提供しており、プログラミング制約の少ないことから、SL4A[9] に Python2.6.2 を実装した。SP 用アプリケーションは、PC 上に Python 環境を構築し開発し、USB 経由で SP に転送した。また、センサのコントロールに AndroidFacadeAPI[10] を一部使用した。

3．PC でのコード開発

微小物の数量測定システム[11] や実長計測システム[12] 用に開発したデータ解析用コードやマッピングコードが利用できること、Wi-Fi 通信機能やメール機能が解析しやすく、独自のコードを組み込みやすいことから、PC でのデータ集計用および表示用コードの開発は VB で行った。また、収集した環境情報をマップ上にリアルタイムで表示するために、多くの有効な API が公開されている GoogleMap[13] を使用し、GoogleMap 上に GPS 情報やセンサ情報を示すマーカは、HTML で記述し VB で制御した。

§5．使用した μP と使用センサ

試作の結果、本稿では次の μP とセンサを使用した。

- Arduino チップ：ATMEGA328P-PU
- 温度センサ：LM35DZ　　・気圧：BMP085
- 湿度：HIH-4030
- GPS モジュール：UltimateV3
- 磁気センサ：MAG3110

§6．通 信 方 法

1．Arduino-SP 間通信

環境センサの設置場所の自由度を上げること、複数の場所に設置した環境センサからのデータを 1 台の SP で同時受信を行うことを考慮し、Arduino と SP 間はケーブル接続とはせずに、短距離通信機能 Wi-Fi とした。Wi-Fi 機器は、XBee[14]、BT module[15]、Host BT[16]、Ethernet モジュール[17] などの Wi-Fi ユニットや通信ユニットをテストし

た結果、RunningElectronics 製 SBDBT[18] とした。

SBDBT には、SPP サーバ機能が搭載されているので、Arduino との通信が容易であり、SP に標準搭載されている Wi-Fi 機能にも特別な設定をすることもなく接続でき、認識率と通信の安定性も高いので、Arduino への負担の軽減とシステムの高速化が期待できる。さらに、小型であるので、システム全体の小型化に有利である。

実装は、SBDBT に SP とのペアリングにより Bluetooth 通信をするためのアダプタ(ドングル)を SBDBT に装着し、Arduino とシリアル通信ポートの Tx と Rx に接続をした。なお、ドングルは Planex 製 BT-MicroEDR1X（通信規格：Bluetooth ver2.1+ EDR）を使用した。

2．PC-SP 間通信

PC と SP 間の通信は、au(KDDI)公衆回線の LTE と 3G を使用した。E-mail の送受信テストは、G-mail、Yahoo-mail、E-mail、C-mail(KDDI)、JavaMail を組み込んだプロトタイプコードで行った。その結果、送受信ポートやアドレスの情報が得やすいこと、VB との連携性や送受信環境の自由度が高いこと、任意の設定が送受信コードに組み込めること、さらに IMAP による G-mail が、本システムとの汎用性と連携が高く、詳細設定も可能なことから、PC と SP 間の長距離通信は G-mail とした。

§7．システム仕様

以上の結果、本システムの仕様は次のように決定した。

§8．システムフロー

1．環境センサによる環境量の計測とデータ送信の処理

Arduino による環境量の測定と SP へのデータの送信は次のフローで行った。

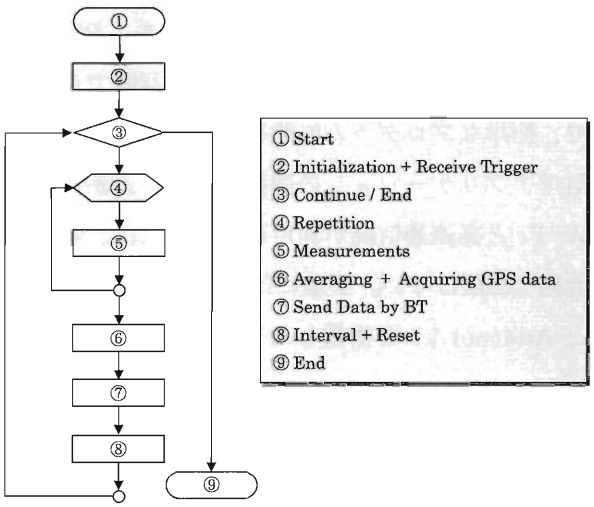

図3　環境測定部の処理システム

システム起動用のトリガーコードを受信し、認証コードであれば計測を開始し、終了コードであれば計測を完了する。計測回数はセンサの自己発熱による誤差を最小にし、かつ精度を上げるために10回/サイクルとした。10回の計測後、1秒間の間隔をあけ平均値を算出した。なお、GPS モジュールを組み込んだ実験機 2 では、10回の計測後、GPS による経緯度の測定のために60秒間 GPS を作動させ、経緯度、高度、方位、時間を確定した。その後、計測データと環境センサ番号を

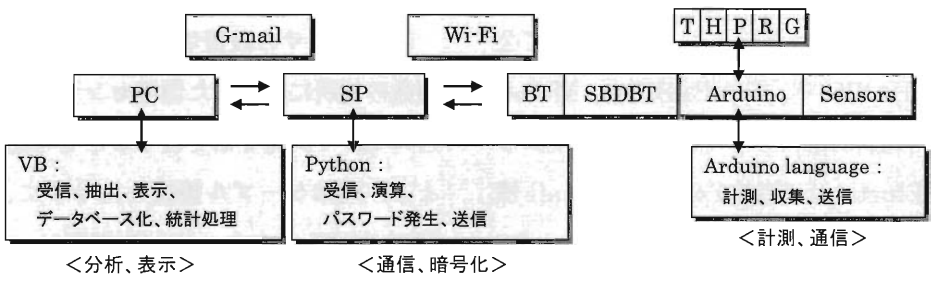

図2　環境計測システム図

Arduino のシリアルポートから SBDBT に出力し、BT ドングルとペアリングしている SP へ送信した。

〈計測用スケッチ〉

圧力センサ：BMP085 の回路及び気圧と高度算出用のコードは、SparkFun 社の datasheet を参考にした[19]。温度センサ：LM35DZ による温度計測用のコードの作成には、NationalSemiconductor 社の datasheet を参考にした[20]。湿度センサ：HIH-4030 による湿度計測用コードの作成には、Honeywell 社の datasheet を参考にした[21]。磁気センサ：MAG3110 による磁気測定用コードは、Freescale Semiconductor 社の datasheet を参考にした[22]。GPS モジュール：Adafruit Ultimate 66cH10Hz GPS モジュール Version 3 による経緯度の測定用コードの作成には、Adafruit 社のチュートリアルを参考にした[23]。Bluetooth モジュール：SBDBT による BT 送信用のコードは、Running Electronics 社のマニュアルを参考にした[24]。

2．SP における処理

SP では、BT 受信部、データ整形部、G-mail 転送部を開発した。BT 受信部では、システム起動後に、Python の初期化、BT 接続のための初期化、G-mail の送信先アドレス、IMAP の設定、使用するポート番号を行った後、BT の接続を開始する。データ整形部では、ペアリングをした Arduino からのデータを受信した場合に、実験機 1 では GPS を起動し、60秒間の位置確定時間後に経緯度の測定を開始し、GPS 情報から経緯度と時間を抽出する。また、実験機 2 では、Arduino から送信された GPS 情報から経緯度と時刻を抽出する。その後、G-mail 転送部において、読み取り用パスワードを発生させ、パスワード、環境センサ番号、経緯度、温度、湿度、気圧、高度からなるデータを G-mail 送信用データへ加工し、G-mail サーバに送信した。

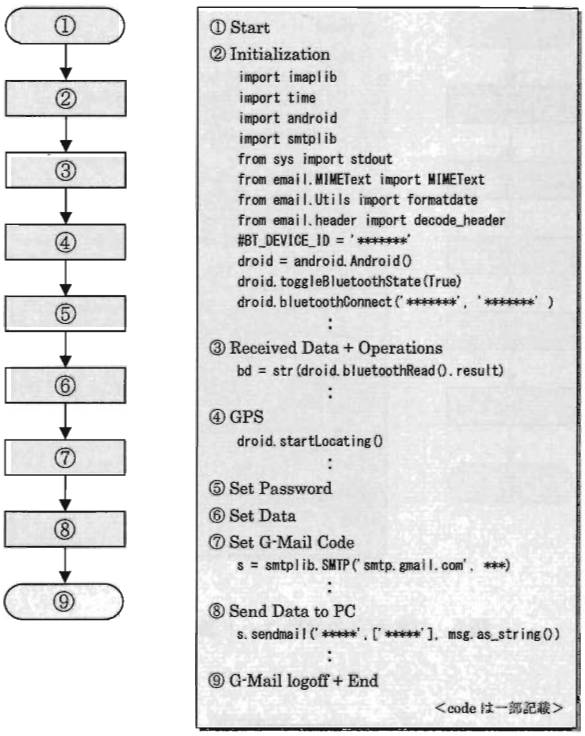

図 4　SP での処理システム

3．VB における処理

PC から SP へ Gmail SMTP Server を経由して、システム起動コードをメール送信するために、Microsoft CDO for windows 2000 Library[26] をリファレンスとして導入した。

システム起動直後の初期設定では、Gmail SMTP Server 設定、送信メールアドレスの設定、プロトコルの設定、SMTP 認証、ポートの設定を行った。

Gmail IMAP4 server からメールを受信するためのリファレンスとして EAGetMail[25] を導入した。システム起動直後の初期設定では、Gmail IMAP Server の設定、受信メールアドレスの設定、プロトコルの設定、データ保護のための SSL 設定と使用するポートの設定を行った。

SP から送信された G-mail の受信タイムラグを極小にするために、タイマー設定により60秒間隔で新着メールの確認を行った。未読メールの中から新着メールを取り出した後、パスワードと認証コードにより環境センサから送信されたメールを選別した。さらに、マッピング表示するメールのシリアル番号と環境センサ番号を選択することに

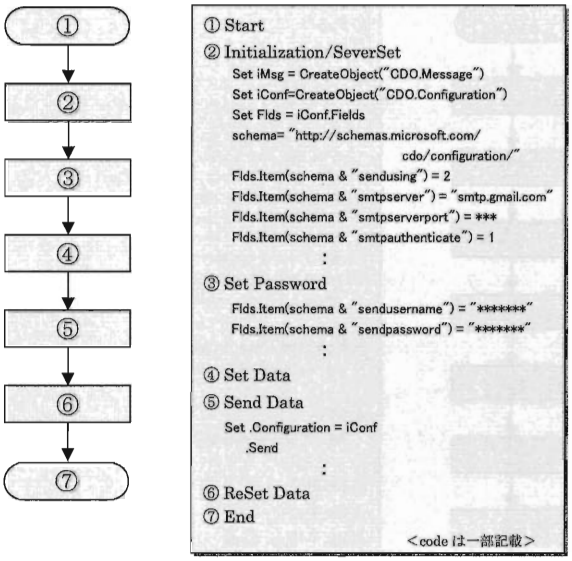

図5　VB からの起動コード送信システム

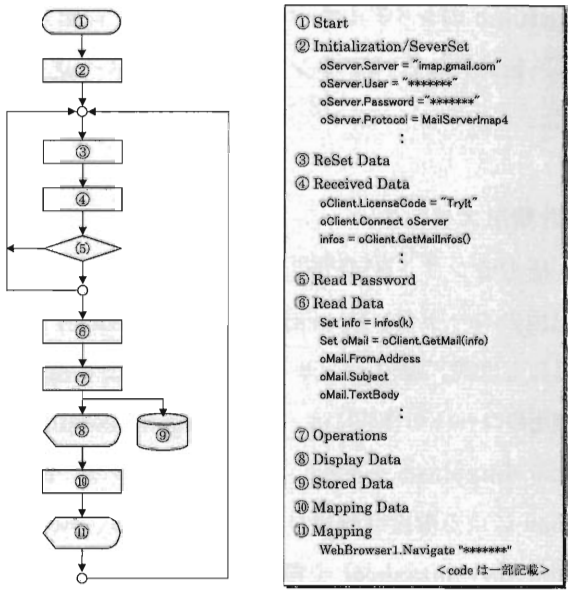

図6　VB での受信処理システム

より、メールデータから経緯度、温度、湿度、気圧、高度、方位、時間を抽出し、マッピング用のデータに加工し、GoogleMap 上にマッピングした。表示位置には、4 つまでのフラッグを準備した。以降、データを受信するごとに、表示内容の更新とマップ表示の更新を行い、同時に、HDD へ出力しデータベース化した。開発した機能として、①環境データのリアルタイム表示、②同データの Map 表示、③環境システムの選択、④履歴表示、⑤表示データの選択、⑥データベース化機能、⑦ trace & logger 機能などである。

§9．実　測

4 台の環境センサと SP による計測実験を行った。図中の吹き出しは、観測場所を示すフラッグであり、クリックにより環境データが表示される。

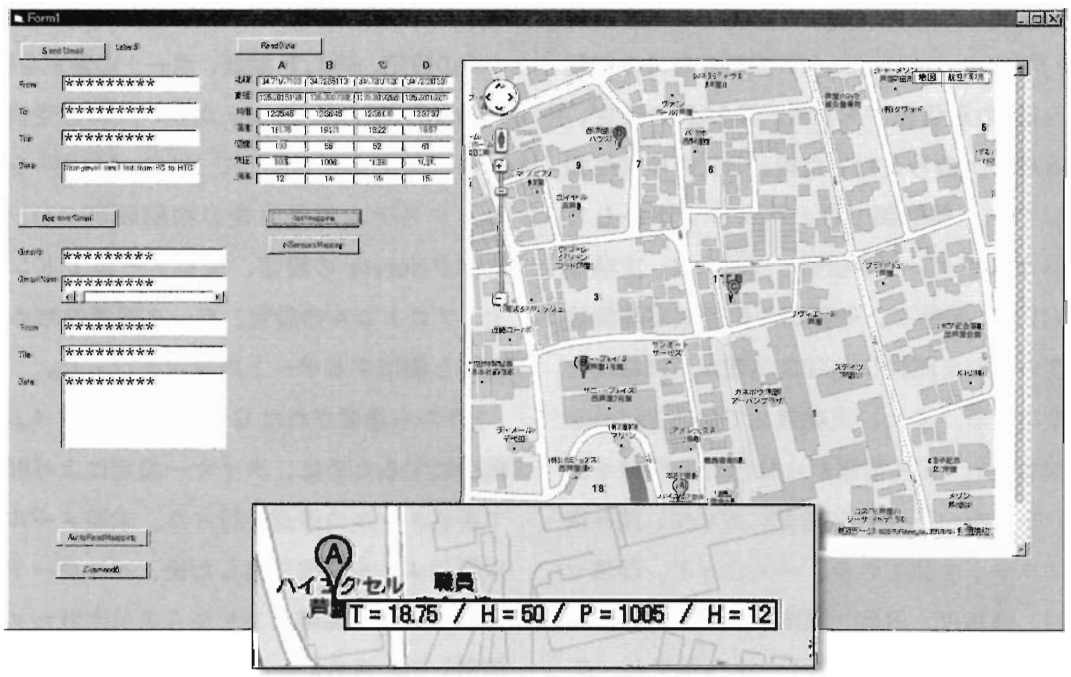

図7　環境計測システムインターフェイス

下図は、その結果であるが、場所により最大で4.5分ほどの時間差が生じていた。システムに公衆ネットワークと一部 Wi-Fi スポットを利用していることが原因と考えられるが、リアルタイムでの表示を目標としているので、タイムラグの要因を調べた。

§10. 問　題　点

環境システムの問題点は、次の 3 点である。

1 ：応答時間

2 ：Net タイムラグ

3 ：バッテリ保持時間

〈実験 1 ：応答時間〉

応答時間 τ は、環境センサが計測開始 trigger を受信してから、測定結果を送信するまでの時間であり、τ_1 をセンサの計測時間とすると、

$$\text{応答時間 } \tau = \text{計測時間 } \tau_1 + \text{GPS 計測時間 } \tau_2 + \text{演算時間 } \tau_3 + \text{Wi-Fi 転送時間 } \tau_4$$

τ_2 は GPS モジュールが、GPS 衛星からの GPS 信号を受信してから経緯度を確定するまでの経緯度計測時間であり、正確な位置決定には、4 つ以上の衛星からの情報を取得し、位置計算が必要になる。そこで、本稿でのプログラムでは、電子透かし組み込み[27]に使用した GPS での位置確定までの時間を参考にし、計測時間を60秒間に設定した。また、各プロセスでの所要時間を計測したところ、τ_1、τ_3、τ_4は～10msec であったので、GPS 計測時間 ≫ 計測時間、演算時間、 Wi-Fi 転送時間となり、応答時間は GPS 計測時間 τ_2 に依存することが分かった。

GPS 計測時間を最適化しシステムを改良するために、本稿で使用した GPS を単独に起動し、経緯度決定までの時間を計測した。GPS での経緯度計測では、上空にある複数の GPS 衛星からの情報を受信して、位置決めを行うので、上空を通過する GPS 衛星数や測定環境、天候の影響を受けると考えられる。そこで、上空に遮るものがない場所で、2013年 4 月～ 5 月における晴天時、曇天、降雨時に各50回測定したところ、全平均が8.1±2.3秒であり、最大は降雨時の15.1秒であった。

表 2　GPS 受信時間

	晴　天	曇　天	降　雨
計測時間（sec）	6.3±0.59	7.6±1.7	12.6±6.8 (max：15.1)
受信可能衛星数	12.3±0.82	9.4±0.88	6.8±0.42

この結果から、本システムでの GPS による経緯度計測に必要な最適待ち時間を20秒としプログラムの最適化を行った。

〈実験 2 ：Net タイムラグ〉

Net タイムラグは、PC-SP 間の公衆回線での遅延と SP-Arduino 間の Wi-Fi 通信時間で生じるが、Wi-Fi 通信に要する時間は ～ msec であったことから、Net タイムラグを PC-SP 通信時におけるタイムラグとし、これを計測した。

公衆回線を利用しているので、回線の混み具合は、時間帯や曜日などに大きく影響を受けると考えられる。au での関西地区におけるトラフィック状況が公開されていないので詳細は不明であるが、「ぷらら」のトラフィックモニタの結果[28]では、曜日による依存性よりも時間帯での依存性が高く、22～24時に回線の利用率は最大になり、 4 ～ 8 時に最小になりその差は約 3 倍であった。そこで、曜日は考慮せず、2013年 4 月～ 5 月に次の 3 つの時間帯で、約 5 分間隔で通信時間を測定した。その結果、全平均が75±57秒となり、通信時間に大きな差を確認した。

表 3　通信時間

	5～7 時	14～16時	22～24時
通信時間（sec）	86±54	78±64	74±59
Max	161	162	155

そこで、階級値を20秒とする通信時間のヒストグラムを作成すると次のようになった。

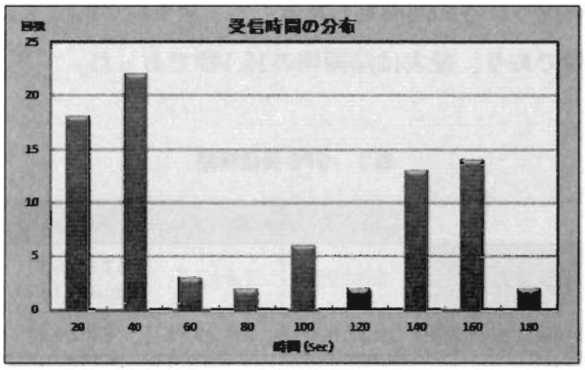

図8　受信に要した時間の分布

20～40秒の短時間で送信できる場合と140～160秒を要する場合があった。今回の実験だけでは、この詳細を解明することは難しいが、おそらくLTE接続と3G接続に因るものと考えられる。そのため、好条件での接続の場合10秒以内で通信が可能であるが、最大120秒～160秒ほどのタイムラグがネットワーク使用により生じることになる。

〈実験3：バッテリ保持時間〉

最適化後の計測システムは、可搬性を重視し電池駆動としたので、連続計測ができる時間に制限がある。そこで、各種センサ（温度、湿度、気圧、高度）、ドングル、SBDBTを装着した状態で、電池の両端電圧を10分間隔で測定し、BT送信が停止するまでの時間を決定した。なお、本システムでは、フル充電したNiMH電池2本（1.40V×2）を、スイッチサイエンス製SFE-PRT-08249[29]を使用し5Vに昇圧して使用した。また、電池の両端電圧は、A/Dコンバートした後、PCで計測した。

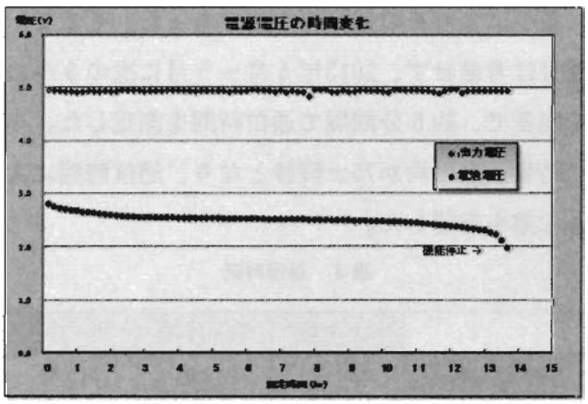

図9　電源電圧の時間変化

図が示すように、システム起動後から電池両端の電圧は徐々に低下し、1.96±0.05Vまで13時間30分±20分間の連続計測が可能であった。また、システムが停止するまでの出力電圧はほぼ一定で、4.94±0.02Vで安定していた。

§11. ま と め

本稿において、4つのセンサを搭載したμPとネットワーク接続したPC+SPシステムからなる遠隔操作が可能な環境計測システムを試作した。本システムは、公衆回線を利用することにより、安価で計測からデータベース化までを自動化できる環境計測システムであり、4つの環境システムにおいて計測開始から表示までの最短時間は、最適化により約30秒であり、連続使用時間は13時間であった。

現状では、Arduinoのアナログ入力端子の制約上、5つ以上のセンサの搭載が困難なため、今後、実装サイズの最小化とともに、搭載センサの増設とPCからの選択機能、連続使用時間の延長方法などを検討したい。同時に、RaspberryPiやPSocなどの新しいμPとSPの組み合わせを検討するとともに、センサ部の開発により、花粉、二酸化炭素、PM2.5なども観測対象とできる計測システムの構築を目指す。また、通信機能を有するTabletへの組み込みにより、Arduino単独で機能するシステム構築を目指す。

参考文献

(1)「環境省大気汚染物質広域監視システム（そらまめくん）」〈http://soramame.taiki.go.jp/〉(20130710)
(2)「環境省花粉観測システム（はなこさん）」〈http://kafun.taiki.go.jp/〉(20130710)
(3)「BASIC Stamp」〈http://www.engr.sjsu.edu/bjfurman/courses/

ME106/ME106pdf/BSprogmanBS2.pdf〉(20130515)
(4) 「Propeller」
〈http://elmicro.com/files/parallax/propellerdatasheet-v12.pdf〉(20130610)
(5) 「BASIC!」
〈https://play.google.com/store/apps/details?id=and.bas&hl=ja〉(20130615)
(6) 「FkmBASIC」
〈http://www.1mobile.com/fkm-basic-582444.html〉(20130620)
(7) 「MobileBASIC」
〈https://play.google.com/store/apps/details?id=com.mobilebasic.FullVersion&hl=ja〉(20130420)
(8) 「Python」
〈http://android.keicode.com/devenv/sl4a-python-install.php〉(20121010)
(9) 「SL4A」
〈http://android.keicode.com/devenv/sl4a-hello-world.php〉(20121010)
〈http://code.google.com/p/python-for-android/〉
(10) 「AndroidFacadeAPI」
〈http://code.google.com/p/android-scripting/wiki/AndroidFacadeAPI〉(20121010)
(11) 黒田正治郎、「無定形・透明・微小物の数量測定システム」、近畿大学短大論集43、2010、pp.19-27
(12) 黒田正治郎、「2焦点距離撮影による実長計測システムの開発」、近畿大学短大論集45、2012、pp.55-66
(13) 「GoogleMap」
〈https://developers.google.com/maps/documentation/javascript/?hl=ja〉(20121101)
(14) 「XBee」
〈http://www.maroon.dti.ne.jp/koten-kairo/works/dsPIC/xbee2.html〉(20130601)
(15) 「BTmodule」
〈http://arduino.cc/en/Main/ArduinoBoardBT?from=Main.ArduinoBoardBluetooth〉(20130605)
(16) 「Host BT」
〈http://www.hobbytronics.co.uk/bluetooth-module〉(20130610)
(17) 「Ethernet モジュール」
〈http://arduino.cc/en/Main/ArduinoBoardEthernet〉(20130610)
(18) 「SBDBT」
〈http://runningele.web.fc2.com/sbdbt/SBDBT-um.pdf (20130701)
(19) 「BMP085 マニュアル」
〈https://www.sparkfun.com/products/retired/9694〉(20130703)
(20) 「LM35DZ マニュアル」
〈http://akizukidenshi.com/download/LM35DZ.pdf〉(20130305)
(21) 「HIH-4030 マニュアル」
〈https://www.sparkfun.com/datasheets/Sensors/Weather/SEN-09569-HIH-4030-datasheet.pdf〉(20130305)
(22) 「MAG3110 マニュアル」
〈http://www.switch-science.com/catalog/734/〉(20130305)
(23) 「GPS モジュールマニュアル」
〈http://learn.adafruit.com/adafruit-ultimate-gps〉(20130305)
(24) 「SBDBT マニュアル」
〈http://www.runele.com/ca1/2/p-r-s/〉(20130719)
(25) 「EAGetMail」
〈http://www.emailarchitect.net/eagetmail/〉(20130401)
〈http://download.cnet.com/EAGetMail-POP3-IMAP4-Component-for-NET/3000-2070_4-10563910.html〉(20130401)
(26) 「Microsoft CDO for windows 2000 Library」
〈http://www.atmarkit.co.jp/fwin2k/win2ktips/428wshmail/wshmail.html〉(20130401)
(27) 黒田正治郎、「A New Techique of Digital Watermark」、200704、第6回電子すかし研究会
(28) 「ぷららのトラフィックモニタ」
〈http://qos.plala.or.jp/traffic/flets/〉(20130401)
(29) 「スイッチサイエンス製 SFE-PRT-08249」
〈http://www.switch-science.com/catalog/78/〉(20130305)
(30) CQ出版、『電脳でちょっと未来を作る』、CQ出版、2012、pp.4-175
(31) CQ出版、『SmartPhoneWorld 3』、CQ出版、2012、pp.12-129
(32) CQ出版、『インターフェイス』、CQ出版、2012、pp.20-121
(33) 牧野浩二、『Arduino電子工作』、東京電機大学出版、2012、pp.1-148
(34) 杉本吉章、『JavaScript入門講座』、技術評論社、2011、pp.229-294
(35) 加藤誠、『日経ソフトウェア』、日経、2013、pp.84-93
(36) 秀和システム、『Perl/CGI大全』、秀和システム、200602、1-170、pp.796-826