

Android OS を用いたフィジカル・コンピューティングの技術習得について

宮崎大学 工学部教育研究支援技術センター

○相川 勝, 甲斐 崇浩, 森 圭史朗, 西岡 祐介, 高塚 佳代子

1. はじめに

Android は、スマートフォンやタブレットなどの OS として急速に普及している。オープンソースの OS であることが普及している要因の一つとなっており、現在ではスマートフォン等の OS としてだけでなく、スマートウォッチやリストバンドなどのウェアラブル端末にまで利用が広がっている。また、Android 搭載のスマート家電としてもルームエアコン、冷蔵庫などが製品化されている。産業分野の組み込み機器にも Android を搭載する動きがあり、今後の普及が期待されている。

情報処理技術班では、情報分野において現在トレンドである技術、今後発展が見込まれる技術のなかで、大学の教育、研究、管理運営の支援業務に活かすことができる技術をキャッチアップし、班（グループ）として共有の技術基盤を構築することを目指している。その一つとしてこの Android OS に焦点をあて、Android 端末と各種ハードウェアが連携することで、人間とコンピュータの意思疎通の方法を拡張する「フィジカル・コンピューティング」の実現を目指して技術研修を行っている。ここでは、平成 25 年度に行った技術研修の活動に関して報告する。

キーワード：Android, フィジカル・コンピューティング, Arduino, ADK, Bluetooth

2. 目的

本研修の目的は、Android OS を活用してフィジカル・コンピューティングを実現（実装）するために必要な知識と技術を習得することである。ひとつの技術や知識を深化させるのではなく、より多くの技術要素の基礎的な部分を習得し、それぞれの技術の繋がりを把握することに重点をおいた研修を行う。これにより今後の技術習得の展開をよりスムーズに進めることが可能になると考える。H25 年度の研修で基礎的な技術習得を目指した具体的なフィジカル・コンピューティング関連の技術要素は以下のとおりである。

- (1) Android アプリケーションを作成できる技術
- (2) Arduino マイコンによるハードウェア制御（LED、スイッチ、センサー、モーター等）ができる技術
- (3) Android ADK（Accessory Development Kit）を用いて Android 端末と Arduino マイコンを連携させる技術
- (4) Bluetooth を介して Android 端末と Arduino マイコンを遠隔制御できる技術

3. 研修プログラム

研修プログラムは、第 1 回から 7 回まで Android アプリケーション作成、第 8・9 回が Arduino マイコンによるハードウェア制御、第 10・11 回が Android ADK を用いた Android 端末と Arduino マイコンの連携、第 12・13 回が Bluetooth を介した Android 端末と Arduino マイコンの遠隔制御を行い、最後に研修プログラムの総まとめとしてすべての技術要素を活用したグループ課題演習を第 14・15 回として実施した。

3.1. Android アプリケーション作成

Android アプリケーション作成では、Android 開発環境として JDK (Java Development Kit)、Android SDK、Eclipse、ADT (Android Development Tools) のセットアップ¹⁾を行い、Android の基本要素である Activity ライフサイクル、イベント処理、レイアウト作成について学習し、Intent による画面遷移、View 設計（リストビュー、ダイ

アログ)、SQLite データベースを用いたデータ保存（データアクセス）のプログラミングを行った。今回の研修では、タブレットとして ICONIA TAB A100 と Nexus7 の 2 機種を、スマートフォンとして XPERIA A (SO-04E) を Android 端末として用いた。各 Android 端末のスペックは表 1 のとおりである。Android 端末にはカメラや各種センサーなど多くのデバイスが搭載されており、Android フレームワークにはこれらのデバイスを使用するための API が準備されているのが特徴である。今回は、使用したすべての機種に搭載されているデバイスとしてカメラおよび加速度センサーを利用したアプリケーション作成を行った。

表 1 Android 端末のスペック

	ICONIA TAB A100	Nexus7	XPERIA A (SO-04E)
メーカー	Acer	ASUS	SONY
Android OS	Android 3.2.1	Android 4.1	Android4.2.2
CPU	Tegra 2 DualCore 1GHz	Tegra 3 QuadCore 1.3GHz	Snapdragon S4 QuadCore 1.5GHz
Memory	RAM:1GB, ROM:16GB	RAM:1GB, ROM:16GB	RAM:2GB, ROM:32GB

● 加速度センサーを用いたアプリケーション

Android の加速度センサーは、重力加速度 ($9.8m/s^2$) に加速度を加えた値を計測し、X、Y、Z の 3 軸ベクトルで取得する。3 軸ベクトルの表現は、Android 端末の背面カメラを表にして机の上においた状態を基準としており、画面を表にした場合は図 1 のような表現となる。

UI スレッドとは独立して動作するスレッドで高速描画を可能とする SurfaceView をグラフィック描画 View として使用し、画面上にボールをイメージした円を描画して Android 端末を傾けると、まるで画面上にボールがあるかのように傾けた方向にボールが転がるアプリケーションを作成した (図 2)。これは、加速度センサーの X 軸と Y 軸の値を取得し、円 (ボール) の位置 (中心座標) を変更し、再描画することで実現している。

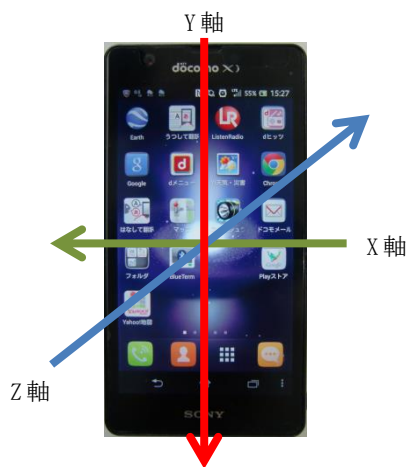


図 1 画面を表にした加速度の 3 軸ベクトル



図 2 画面を転がるボールアプリケーション

3.2. Arduino マイコンによるハードウェア制御

Arduino は、接続されたスイッチやセンサーの値を簡単に読み取ることができ、そして LED やモーターを制御できるマイコンボードである。Arduino はハードウェアとソフトウェアのすべての仕様が公開されているオープンソース・ハードウェアである。今回の研修で使用した Arduino マイコンボードは、Arduino Uno R3 である。Arduino の総合開発環境である「Arduino IDE」をセットアップ²⁾し、USB ケーブルで Arduino マイコンと接続してプログラムを作成 (スケッチ作成という) し、Arduino マイコンに書き込むことによってプログラムを Arduino マイコン上で動作させることができる。

今回の研修では、LED、タクトスイッチ、ボリューム、赤外線距離センサー、超音波距離センサー、DC モーターを制御するプログラムについて基礎的な学習を行った後に、これらを組み合わせたプログラムとして「セ

ンサーでモーターを制御するプログラム」を演習課題として実施した。具体的には、距離センサー（赤外線距離センサー、超音波距離センサー）で計測した距離によって、DC モーターの速度を制御するプログラムであり、プログラムの仕様は次のとおりである。

● センサーでモーターを制御するプログラム

- (1) 距離センサー（赤外線距離センサー、超音波距離センサー）で距離を計測し、対象物からある一定の距離以上離れている場合（安全距離）は、指定の回転速度でDC モーターを回転（正転）させる。
- (2) 対象物との距離が近づくにつれて、DC モーターの回転速度を低下させ、限界値（危険距離）以下になった場合、DC モーターの回転を停止する。
- (3) 距離センサーで計測した距離と DC モーターの回転速度の設定値を表 2 に示す。また、見た目でも判断できるように 4 つの LED を使用し、各区間で点灯する数を設定する。危険距離の場合は 4 つすべて点灯し、安全距離の場合は 4 つすべて消灯する。各区間で点灯する LED の数も表 2 に示す。

表 2 対象物との距離と DC モーターの回転速度

対象物との距離	DC モーターの回転速度	点灯する LED の数
0cm ~ 5cm 以下 (危険距離)	0	4
5cm ~ 15cm 以下	50	3
15cm ~ 30cm 以下	100	2
30cm ~ 50cm 以下	150	1
50cm ~ (安全距離)	200	0

回路図を図 3 に示す。距離センサーには、赤外線距離センサーか超音波距離センサーのどちらかを使用する。DC モーターを回転させるためのモータードライバ IC として東芝製の TA7291P を 1 つ、DC モーターを 1 つ、LED を 4 つ、抵抗 (330Ω) を 4 つ使用する。実装図は図 4 のとおりである。

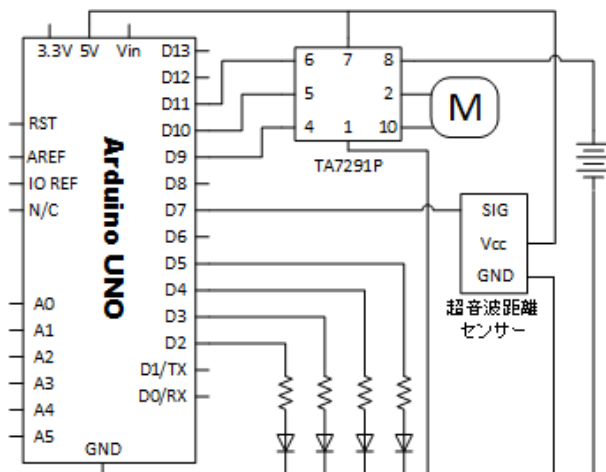


図 3 センサーによる DC モーター制御の回路図

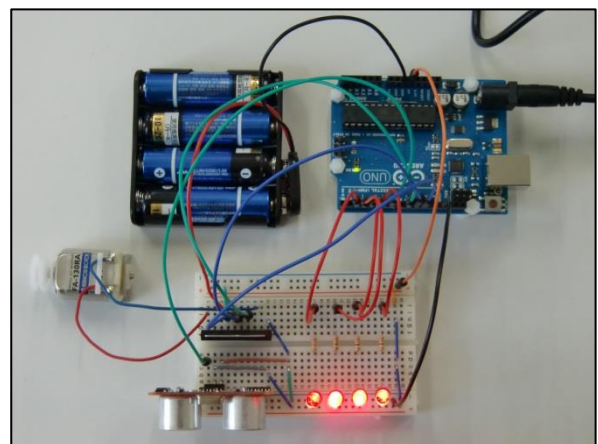


図 4 センサーによる DC モーター制御の実装図

3.3. Android ADK を用いた Android 端末と Arduino マイコンの連携

AOA (Android Open Accessory) は、USB インターフェイスを介して Android 端末と外部デバイスを接続するためのインターフェイス仕様である。この AOA の開発キットが、Android ADK (Accessory Development Kit) である³⁾。ADK を利用すれば Android 端末と外部デバイスを USB インターフェイスで接続してデータのやり取りができる。具体的には ADK 仕様に対応した Arduino マイコンボードと、Arduino IDE に ADK をセットアッ

プして外部デバイスを開発する。

ADK は、通常の USB 機器接続の形態とは逆の構成となり、Android 端末側が「USB デバイス」、ADK 対応 USB 周辺機器が「USB ホスト」という接続形態になる。よって、USB ホストを備えていない Android 端末でも USB インターフェイスを使用して、ADK 対応の USB 周辺機器と接続が可能である。また、電源供給についても逆となり、ADK 対応 USB 周辺機器側が、Android 端末に対して 5V の電源を供給することになる。これは、Android 端末のバッテリー問題を解決するための手段となる。ADK 対応の USB 周辺機器は「USB アクセサリ」と呼ばれる。

今回は USB アクセサリとなる ADK 対応の Arduino マイコンボードとして、「Arduino Mega ADK」を使用し、Android 端末と Arduino マイコンを USB で接続してハードウェアを制御するプログラムを学習した。

● USB アクセサリモードの Android アプリケーション側通信手順

USB アクセサリモードの Android アプリケーションの通信手順を図 5 に示す。まず、onCreate メソッド内で①USB マネージャを取得し、②USB アクセサリの挿抜通知受取準備(ブロードキャストレシーバ登録)を行う。onResume()メソッド内で、③USB アクセサリの取得と権限チェックを行い、④USB アクセサリオープン処理で入出力用のストリームを取得し、⑤取得した出力ストリームで USB アクセサリ側へデータを送信し、入力ストリームで USB アクセサリ側からのデータを受信する。通信が終了した場合は、⑥USB アクセサリクローズ処理を行う。

● Android 端末から Arduino へアナログ値出力による DC モーター制御

Android 端末のシークバーの操作で、Arduino ADK マイコンに接続されている DC モーターの速度を制御するプログラムを作成した。シークバーは、0~255 の値を表現し、シークバーの値が変化する度に、Android 端末から USB を介して Arduino ADK マイコンにデータを送信し、Arduino ADK マイコン側で受信した値によって、DC モーターの速度制御を行う (図 6)。

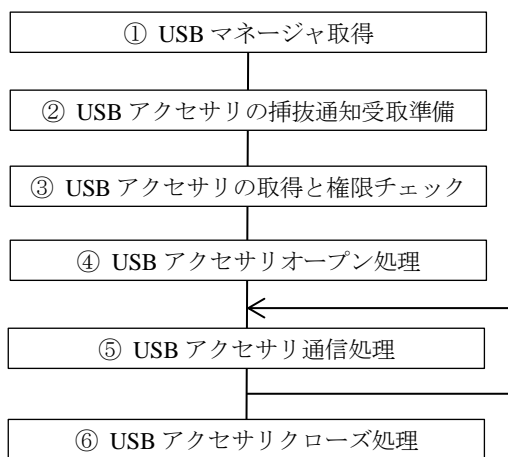


図 5 USB アクセサリモードの Android アプリケーション通信手順

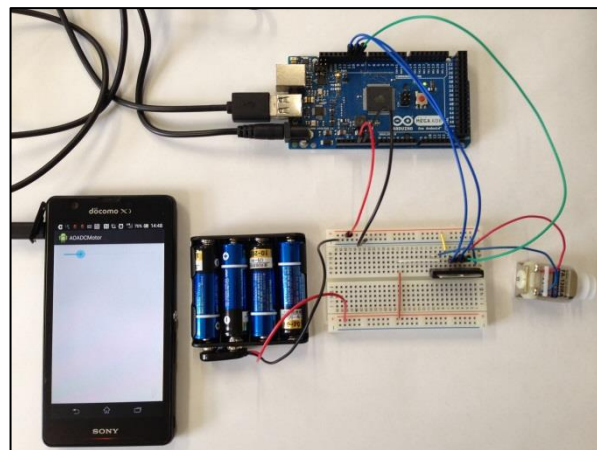


図 6 Android 端末から ADK による DC モーター制御

3.4. Bluetooth を介した Android 端末と Arduino マイコンの遠隔制御

ADK は USB アクセサリモードでの通信であり、USB ケーブルによる有線でのデータ通信であった。有線で接続された状態でも利用用途はあるが、無線での通信の方が利用用途やアイデアが格段に広がる。そこで、無線モジュールとして Bluetooth を用いた Android 端末と Arduino マイコンの遠隔制御を学習した。

● Bluetooth の概要

Bluetooth は数 m から数十 m 程度の距離の情報機器間で、電波を使い簡易な情報のやり取りに使用される通信技術であり、PAN (Personal Area Network) を実現することを想定した無線通信規格である⁴⁾。無線 LAN の規格である IEEE802.11b と同じ 2.4GHz 帯を用いており、スマートフォンやタブレットでも、Bluetooth のハードウェアデバイスが多くの機種で搭載されている。

Bluetooth には、電波強度を規定したクラスという概念がある。各機器はいずれかのクラスに分類され、クラスによって通信到達距離が異なる。表 3 は各クラスの出力値と到達距離を表したものであるが、この距離は保証されるものではない。

Bluetooth はその特性上、様々なデバイスでの通信に使用されるため、機器の種類ごとに策定されたプロトコルがあり、それらの使用方法をプロファイルとして標準化している。通信を行う機器同士が同じプロファイルを待つ場合に限り、そのプロファイルの機能を利用した通信が行える。今回は、Bluetooth 機器を仮想シリアルポート化するプロファイルである、SPP (Serial Port Profile) を使用する。

Bluetooth 機器は、それぞれに世界で一意的な機器アドレス (MAC アドレス) を持っている。Bluetooth 機器同士での接続では、最初にそれぞれの MAC アドレスを指定し、パスキーと呼ばれる文字列を入力することで、通信時の通信暗号化に必要な暗号鍵を交換する。この操作をペアリングと呼び、ペアリングが行われた機器同士は接続を確立することができる。

表 3 Bluetooth のクラス

クラス	出力	到達距離
Class1	100mW	100m
Class2	2.5mW	10m
Class3	1mW	1m

● Bluetooth モジュール

今回使用した Bluetooth モジュールは、Robotech 製の「RBT-001 (EasyBT)」と SparkFun 製の「Bluetooth Mate Gold WRL-12580」の 2 つである。それぞれのモジュール仕様は表 4 のとおりであり、プロファイルとして SPP を使用する。

表 4 Bluetooth モジュール

仕様	RBT-001(EasyBT)	Bluetooth Mate Gold (WRL-12580)
メーカー	Robotech	SparkFun
バージョン	Bluetooth 2.0	Bluetooth 2.0
クラス	Class2	Class1
プロファイル	GAP SDAP SPP	GAP SDAP SPP
UART default	9600	115200
パスキー	0000	1234

● Bluetooth 通信 Android アプリケーションの処理手順

Bluetooth API を用いた Bluetooth 通信 Android アプリケーションの処理手順は、はじめに「①Bluetooth サポート有無の確認と有効化」を行い、次に「②Bluetooth デバイスの検索」によって接続する機器 (デバイス) を特定し、そして「③RFCOMM による接続」で機器 (デバイス) との接続および通信を行う手順となる。

「①Bluetooth サポート有無の確認と有効化」では、Android 端末が Bluetooth に対応していることを確認し、対応していた場合には Bluetooth の ON/OFF (有効か否か) を判断し、有効でない場合 (OFF) には有効 (ON) にするようユーザに促す (図 7)。

Bluetooth の使用が有効な場合、接続を行う「②Bluetooth デバイスの選択 (検索)」を行う。ペアリング済のデバイスを選択するか、ペアリングしていないデバイスを検出してから選択するか、どちらかの方法で Bluetooth デバイスを選択する (図 8)。

Android フレームワークでは、BluetoothSocket API によって、Bluetooth の基盤をなすプロトコルである RFCOMM を用いて通信機能を提供する。RFCOMM は、Bluetooth でシリアル通信をエミュレートするプロトコルである。BluetoothSocket は TCP/IP のソケットインターフェイスと似ており、サーバ/クライアントモデルで通信を行う。今回は、クライアント側のみ（Bluetooth モジュールがサーバ側になる）を作成する。選択した Bluetooth デバイスを用いて、SPP のプロファイル ID を指定してソケットを作成し、サーバ側（Bluetooth モジュール）に接続を行う。接続に成功した場合は、ソケットからデータ通信の入出力ストリームを取得し、その入出力ストリームを使用してデータ通信を行う。データ通信のプロトコル定義を自作して、そのプロトコル（コマンド）に沿って通信機器同士（クライアントとサーバ）が協調しながら通信処理を行う。

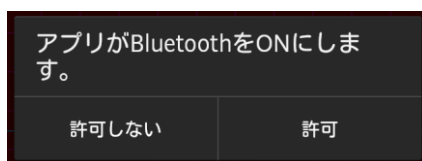


図7 Bluetooth 使用の確認画面



図8 接続するデバイスの選択

3.5. グループ課題演習

研修の最後に今まで学んできた技術を活かし総まとめとしてグループ課題演習に取り組んだ。グループ課題演習のテーマは、「Bluetooth 通信によるリモコンカー製作」である。Android 端末にジョイスティックを模擬したビューを実装し、その操作を Bluetooth 通信によって Arduino を搭載したリモコンカーで受信し、制御するアプリケーションである。Bluetooth 通信によるリモコンカーの概要を図9に示す。

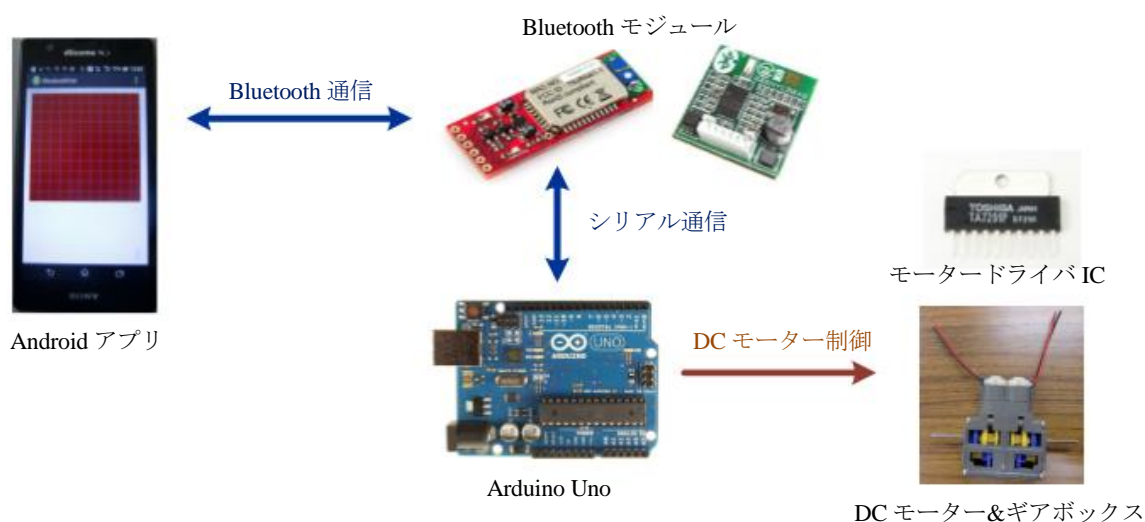


図9 Bluetooth 通信によるリモコンカーの概要

● リモコンカー部と電子回路の製作

ユニバーサルプレートをベースに、後輪駆動として2つの DC モーターとダブルギアボックスで左右のタイヤを独立的に駆動させ、前輪には細かな動きを可能とするためにボールキャビネットを取り付ける（図10）。DC モーターを制御するためのモータードライバ IC（TA7291P）を2つ使用し、モータードライバ IC の電源として、単三電池4本を使用する。また、Arduino への外部電源としては9V 電池を使用する。

DC モーターの回転で前進、後退、加速、減速を制御し、2つの DC モーターの速度差によって左右のハンドル操作を制御とする。Android 端末と Bluetooth 通信を行う Bluetooth モジュールとして、「RBT-001」と「Bluetooth

「Mate Gold」を用いて電子回路を製作する（図 11）。

● Arduino スケッチの作成

Bluetooth モジュールが受信したデータは、非同期式シリアル通信として Arduino マイコン側に送信される。よって Arduino マイコン側からすれば、Android 端末とのデータのやり取りを非同期式シリアル通信として処理することで実現できる。具体的には、Arduino マイコンの TX ピン、RX ピンを Bluetooth モジュールの RX ピン、TX ピンとクロス接続し、Arduino のシリアル通信関数（Serial.available()、Serial.Read()、Serial.Write()）を用いてデータ通信処理のスケッチを作成する。

駆動部の 2 つの DC モーターを制御するためのコマンドプロトコルは表 5 のように指定する。左 DC モーターの速度調整には、PWM ピンである 3 番ピンを使用し、回転方向入力として 4 番、5 番ピンを使用する。右 DC モーターの速度調整には、PWM ピンである 9 番ピンを使用し、回転方向入力として 10 番、11 番ピンを使用する。コマンドプロトコルに従って、受信した 4 つのデータ、左 DC モーター回転方向 (1or0)、右 DC モーター回転方向 (1or0)、左 DC モーター速度、右 DC モーター速度によって、前述した出力ピンへ適切な値を出力し、モータードライバ IC を介して DC モーターを制御するスケッチ（プログラム）を作成する。

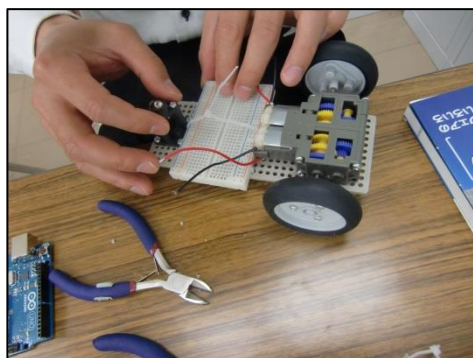


図 10 リモコンカーの製作

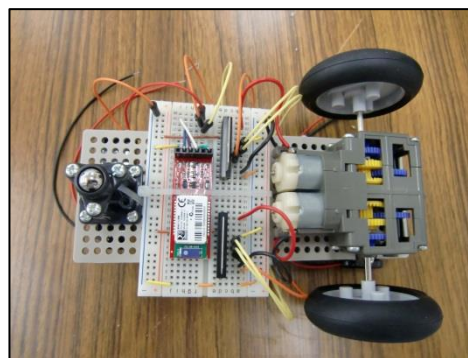


図 11 電子回路の製作

表 5 DC モーター制御プロトコル

1 バイト目	2 バイト目	3 バイト目	4 バイト目	5 バイト目
制御コマンド 0x50	左 DC モーター 回転方向(1or0)	右 DC モーター 回転方向(1or0)	左 DC モーター 速度(0-255)	右 DC モーター 速度(0-255)

● リモコンカーを制御する Android アプリケーションの作成

リモコンカーを制御するコントローラ部としてジョイスティックを模擬した Android アプリケーションを作成する。Android 端末の画面に縦横 11×11 のマス格子を描画する。Android 端末の画面サイズに合わせて 1 つの格子のサイズは異なる。画面をタップするとそのマスを交点として横、縦のマスの色が黄色になり十字を描画する（図 12）。操作としては、11×11 の中心を 0（停止状態）として、Y 軸方向が前進・後退および速度コントロールを行う。上が前進、下が後退となり、上にスライドするほど前進の速度が増加し、下にスライドするほど後退の速度が増加する。X 軸方向は左右のハンドル操作を行う。X 軸の中心を 0 として、右方向にスライドすると右に曲がり、左方向にスライドすると左に曲がる。右に行くほどハンドルを大きく右にきった状態に、左に行くほどハンドルを大きく左にきった状態になる。ハンドル操作は、図 13 のプログラムによって左右の DC モーターの速度差によって実現している。DC モーターは 0～255 の PWM 出力で速度を調整する。例えば、タップしている位置が(X,Y)=(1,4)であった場合、左 DC モーターのスピードは 10 になり、右 DC モーターのスピードは 51 となり、右 DC モーターの方が 41 大きいので左に曲がることになる。逆にタップしている位置が(X,Y)=(9,4)であった場合、左 DC モーターのスピードは 51 になり、右 DC モーターのスピードは 10 となり、左 DC モーターの方が 41 大きいので右に曲がることになる。

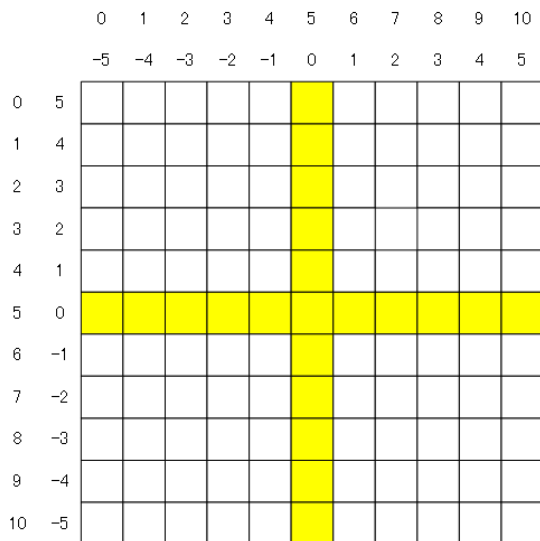


図 12 コントローラ部の Android

```

motorPower_L = convertPower(x, y); //左モーター
motorPower_R = convertPower(10-x, y); //右モーター

int convertPower(int x, int y){
  int unit = 255/5;
  int mapx = -5 + x;
  int mapy = 5 - y;

  if(mapx >= 0)
    return unit * mapy;
  else
    return ((unit*x) /5) * mapy;
}

```

図 13 ハンドル制御プログラム

4. おわりに

今回の研修では、フィジカル・コンピューティングを実現（実装）するために、Android アプリケーション、Arduino マイコンによるハードウェア制御、Android ADK による Android 端末と Arduino マイコンの連携、Bluetooth を介した Android 端末と Arduino マイコンの遠隔制御について研修を行い、それぞれの技術要素における基礎的な知識と技術を習得することができた。研修の最後では、研修で学習した技術要素を活用したグループ課題演習として「Bluetooth 通信によるリモコンカー製作」に取り組んだ。一つ一つの技術要素は基礎的なものであるが、それらを組み合わせることによって実用的なフィジカル・コンピューティングを実装できることを体験することができた。また、グループ研修として取り組んだことによって、班（グループ）で共有する技術基盤のひとつを立ち上げることができたと考える。今後は、それぞれの技術要素に関し、知識と技術力を高めつつ、新しい技術要素としてシングルボード・コンピュータ（BeagleBone Black、Raspberry Pi、BeagleBoard-xM）の技術習得も行き、Android アプリケーション開発から組み込みシステム開発まで、提供技術の幅を広げ、技術基盤の確立を進めていきたいと考えている。

謝辞

本研修は、平成 25 年度学部長裁量経費の助成により実施されました。助成事業を支援して頂きました関係者の方々に厚くお礼申し上げます。

参考文献

- 1) 高見知英, よくわかる Android アプリ開発の教科書 Android 4.2 対応版, マイナビ
- 2) 牧野浩二, たのしくできる Arduino 電子工作, 東京電機大学出版局
- 3) 伊藤 元, Android×Arduino でつくるクラウド連携デバイス, インプレスジャパン
- 4) 邑中雅樹, 組み込み Android プログラミング入門, 技術評論社