組込みオペレーティングシステム実習II用の教材の作成２

電子情報技術科（現：関東職業能力開発大学校）　岡久　潤一

Teaching materials for Embedded Operating System Practice

Junichi OKAHISA

概要　Linux が動作するシングルボードコンピュータ【Raspberry Pi】を使い、Bluetooth で遠隔制御する教材を作成した。

1. はじめに

Raspberry Pi を用いた「組込みオペレーティングシステム実習II」の教材を以前作成し、利用していたが下記の理由で、課題演習部分や一部内容を変更することにした。昨年度から当校でも始まった 1 年生の科目「実践技能者基礎実習[電子情報分野]」で、前回作成した内容の一部を使用したことと、コース制の廃止で Android アプリケーション制作の内容がなくなったこと、Wi-Fi を利用した演習課題を実施していたが、ネットワークの設定や準備の手間がかかることから変更を行った。

追加・変更した内容は、今年度の総合制作において、Bluetooth 経由で操作するラジコンカーの制作に取り組んだので、Bluetooth の環境設定や制御・通信方法等を内容に取り入れ、追加の演習課題としても利用できるよう、まとめることにした。

2. Bluetooth を用いた内容について

2.1 開発環境

Raspberry Pi は、Bluetooth が搭載されている Zero WH を、OS は Raspberry Pi 専用の Raspbian(stretch)を使用した。また、Raspberry Pi 側の制御には、Python3.4 を、Android アプリケーションの制作には、開発環境「Android Studio3.2」を使用した。

2.2 環境設定

Raspberry Pi 同士で通信を行う場合は、追加の設定なしでデータの送受信ができたが、Android タブレットと Raspberry Pi で通信する場合は、以下の設定を行った。

①シリアル通信(SPP)の設定

②Bluetooth 用拡張モジュール「pybluez」のインストール

2.3 Raspberry Pi 同士の通信

2 台の Raspberry Pi での文字列の送受信は、以下の内容で確認した。

①親機側のコントローラを探索可能に設定（図 1）

②親機側（受信）プログラムを実行

③子機側（送信）プログラムを実行

④送受信の確認（図 2、図 3）

親機側のプログラムでは、ソケット生成・登録と接続、他の端末とのデータの送受信を行っている。

子機側では、ソケット生成と接続、他の端末のデータの送受信を行っている。文字列「quit」が送信されると、親機・子機どちらも接続が切断される。

```
pi@raspberrypi:~ $ bluetoothctl -a
[NEW] Controller B8:27:EB:2F:C3:F4 raspberrypi [default]
Agent registered
[bluetooth]# discoverable on
Changing discoverable on succeeded
[CHG] Controller B8:27:EB:2F:C3:F4 Discoverable: yes
[bluetooth]#
```

図 1　コマンドによる探索可能設定（親機側）

図 2　文字列を送信（子機側）

図 3　文字列を受信（親機側）

また、子機側から送信した文字を親機側からそのまま送り返すよう内容を追加した。実行確認までは、上記と同じである。（図 4）

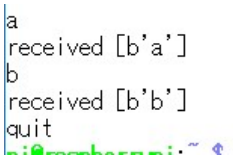

図 4　文字列の送受信（子機側）

2.4 Android App（送信）- Raspberry Pi（受信）

2.4.1　Android アプリケーション（子機側）

子機（Android タブレット）側から文字データを送信し、親機（Raspberry Pi）で、データを受信する。動作確認は、以下の手順で行った。

①親機側のコントローラを探索可能に設定
②タブレット側でのペアリング設定
③親機側で信用登録
④シリアル通信（SPP）の設定
⑤子機側（送信）アプリの作成
⑥Android アプリの実行と送受信の確認

親機側では Raspberry Pi 同士の通信と同様にソケット生成・登録と接続、他の端末とのデータの送受信を行っている。

子機側では、他の端末のスキャンと接続デバイスのペアリング設定、他の端末の接続、データの送受信を行っている。

Android アプリケーションは、Developers ガイドを参考に以下の API（表 1）を使用した。

表 1　使用した API

API	内容
BluetoothAdapter	通信に必要な情報を格納
BluetoothDevice	デバイス情報を格納
BluetoothSocket	ソケット情報を格納
OutPutStream	出力用ストリーム
InputStream	入力用ストリーム

そして、データを受信する Raspberry Pi の MAC アドレスと UUID をソースコード内に指定し、ボタンを押したときの処理や通信部分は Java を用いた。ボタンの配置や画面の作成はレイアウトエディタで作成した。

「AndroidManifest.xml」ファイルに Bluetooth に関係するパーミッションを追加した。

実行時は、子機側の Android アプリを起動後親機側でデータを受信できるように準備する（図 5）。そして、子機側端末で入力した文字列を送信する（図 6）。親機側は直接文字列を受信するため、文字がそのまま画面に直接表示される。（図 7）

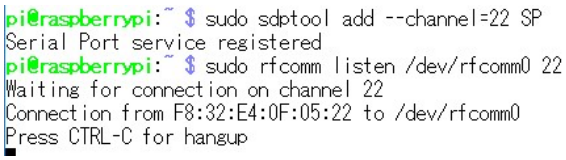

図 5　通信設定画面（親機側）

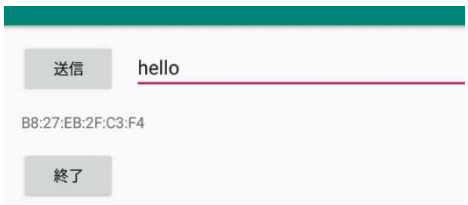

図 6　送信画面（子機側）

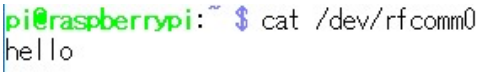

図 7　受信画面（親機側）

2.4.2 Python プログラム（親機側）

2.4.1 の送受信では、毎回接続の設定を行う必要がある。後で GPIO 制御することを考え、通信部分を Python プログラムで作成することにした。

なお、次の手順で設定及び動作の確認をした。

①「pybluez」依存パッケージをインストール
②「pybluez」のインストール
③親機側（受信）プログラムを実行
④子機側（送信）アプリケーションを実行
⑤動作確認

確認は、親機側の Python プログラムを先に実行、受信待機状態にして、子機側の Android アプリを起動する。接続すると、子機側の MAC アドレスが表示される。そして子機側端末で入力した文字列を送信すると、親機側で文字列を受信し、画面に表示される。（図 8）

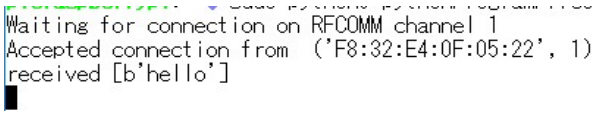

図 8　文字列の受信画面（親機側）

2.4.3 操作ボタンの配置と GPIO 制御

Android アプリの送信画面にボタンを配置し、各ボタンに対応した処理を行えるようにした。

ON／OFF ボタンを追加して Raspberry Pi の GPIO 制御（LED やモータを駆動・停止）ができるようにするだけでなく、終了ボタンで Raspberry Pi のシャットダウン処理もボタンで実行できるようにした。また、Raspberry Pi の起動後に自動で親機側のプログラムが起動するように設定した。

設定及び動作確認は次のように行った。

①「rc.loacl」ファイルに自動実行部分の追記
②親機側（受信）プログラムを実行
③子機側（送信）アプリケーションを実行
④動作確認

2.4.2 と同様に親機側の Python プログラムを先に実行し、子機側の Android アプリを起動する。ON または OFF ボタンを押す（図 9）と文字列が送信され親機側で文字列を受信する（図 10)。

終了ボタンを押すと画面が閉じて Raspberry Pi がシャットダウンする。（図 11）

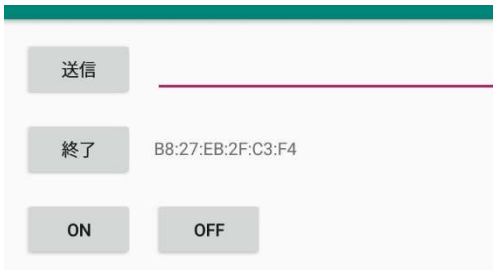

図 9　ON/OFF ボタン（子機側）

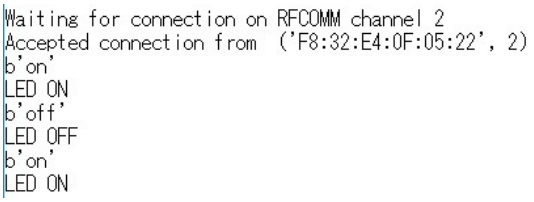

図 10　文字列の受信画面（親機側）

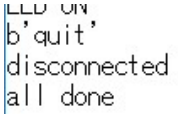

図 11　終了処理

さらに、子機の Android アプリの画面に受信ボタンを配置した（図 12)。親機から送信された文字列を、受信ボタンを押すことで受信できるようにした。

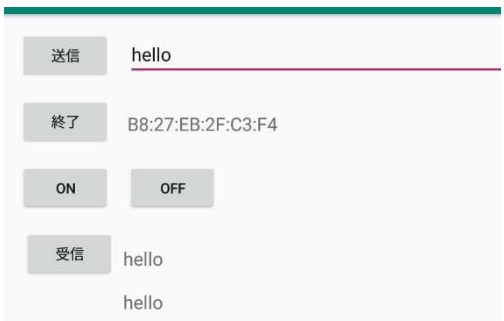

図 12　親機からの受信

2.5　総合課題（Bluetooth ラジコンカーの製作）

これまでの内容や総合制作で取り組んだ内容をもとに、Bluetooth 経由で操作するラジコンカーを制作する内容にした。

ラジコンカー（図 13）は、2 個のステッピングモータで駆動させる。前後に進むだけでなく、右左折できるようにしている。また、距離センサを搭載し、一定の距離まで近付いたら自動で停止するようにしている。動作中でも、停止できるようスレッドを使い並列処理を行っている。

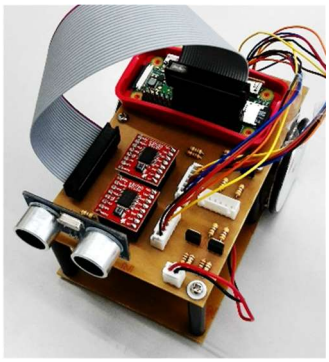

図 13　本体

操作画面（図 14）は、矢印・STOP ボタンにより前後左右の移動と停止を、また、SPEED_UP、DOWN ボタンでスピードの調節を、そしてラジコンカーに 4 つの LED を取り付けた際に、LED の ON/OFF や明るさを調節できるようにした。

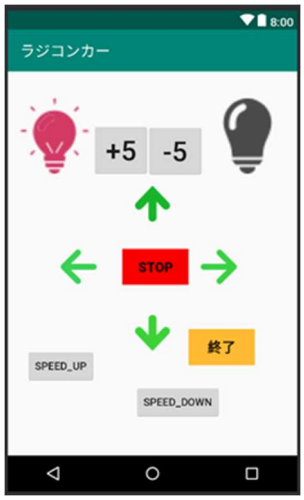

図 14　操作画面

ラジコンカーの起動・動作確認は、次の手順で行う。

①Raspberry Pi を起動

②Android アプリを起動

③操作して動作確認をする。

④終了ボタンを押してアプリと Raspberry Pi を終了する。

3. おわりに

実習の目標としている Linux のコマンド操作や環境・通信の設定の習得だけでなく、今回追加した内容を通して、電子情報技術科の総合的な内容を確認することができると考える。

今後は、BLE（Bluetooth Low Energy）による制御や Android アプリの制作、各種センサを使い、さらに内容を発展させていけるようにしていきたい。

参考文献

1) Python Software Foundation
https://docs.Python.org/ja/3/library/socket.html?highlight=bluetooth#module-socket

2） Pybluez
https://pybluez.github.io/

3) Android Developers ドキュメント ガイド
https://developer.android.com/guide/topics/connectivity/bluetooth?hl=ja