



メカトロニクスコースにおけるプログラミング実験 実習

メタデータ	言語: jpn 出版者: 公開日: 2019-01-28 キーワード (Ja): キーワード (En): 作成者: 平尾, 康起, 和田, 健, 前田, 篤志 メールアドレス: 所属:
URL	https://doi.org/10.24729/00007504

メカトロニクスコースにおけるプログラミング実験実習

平尾康起*, 和田 健**, 前田篤志**

Programming Experiments in Mechatronics Course

Kohki HIRAO*, Takeshi WADA** and Atsushi MAEDA**

要旨

大阪府立大学工業高等専門学校・メカトロニクスコースでは、4年次の「電子機械工学実験Ⅰ」においてプログラミング実験実習を実施している。使用言語に関する導入教育を実施後、C#プログラミングについては「ZigBee 通信を用いたチャットプログラム」と「シリアル通信による自動演奏プログラム」の作成を実験テーマとしている。また、「音声認識プログラム」を作成する Python プログラミングでは、「スマートスピーカーの製作」と「音声によるロボット制御」を課題として提示している。本稿では実験実習の内容および学生アンケートの結果を紹介するとともに、次年度以降の改善点について議論する。

キーワード: C#, ZigBee 通信, シリアル通信, Python, 音声認識

1. はじめに

平成 23 年度の法人化に合わせて設置されたメカトロニクスコースでは、『ものづくり』能力を基盤として持ちながら、産業構造の変革にも対応できる、機械と電気とコンピュータを統合する能力を有する人材の育成」を教育目標に掲げている。順次的なメカトロニクス教育体系を構築するために、カリキュラムには機械、制御、電気電子分野の講義科目をバランスよく配置している。また、メカ、ハード、ソフトをモジュール化するために必須となる技術の融合を目的に、コース独自の実験実習を各学年の必修得科目として展開している。

以下、本稿では、4年次の「電子機械工学実験Ⅰ」で実施しているプログラミング実験実習の内容を紹介するとともに、学生アンケートの結果をもとに次年度以降の改善点を考察する。

2. プログラミング実験実習

2.1 C#プログラミング 第1週では「C#プログラミング」の導入教育を実施している。スライドによる解説の後、教員のプログラミング画面をスクリーンに投影し

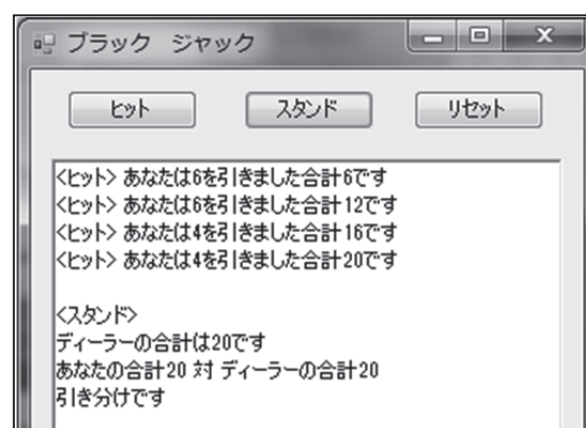


図1 ブラックジャックシミュレーターの実行例

ながら、学生全員が同時進行で「カプセルトイシミュレーター」の作成に取り組む。硬貨を入れレバーを回すとカプセル入り玩具が出てくる通称「ガチャ」のシミュレーターであり、課金アイテムを消費することで毎回異なるキャラクターが得られる。プログラムが正しく実行されていることを確認した後、成績を評価するための課題として「ブラックジャックシミュレーター」の作成を提示している。カードの合計数が 21 に近い方が勝ちとなるトランプゲームであり、プレイヤーは「ヒット」でカードを引き、「スタンド」で引くことを止めてディーラーと合計数を比較する仕様とした（図 1）。評価項目は実行結果とプログラムの読みやすさとし、インデント、意味のある変数名、コメント文等を用いた「良いコード」の書き方を指導している。以上、本実験実習の学習項目は下記のとおりである。

・ Visual Studio、Visual C#

2018 年 8 月 20 日 受理

* 総合工学システム学科 技術支援室

(Dept. of Technological Systems Engineering: Technical Experts)

** 総合工学システム学科 メカトロニクスコース

(Dept. of Technological Systems Engineering: Mechatronics Course)

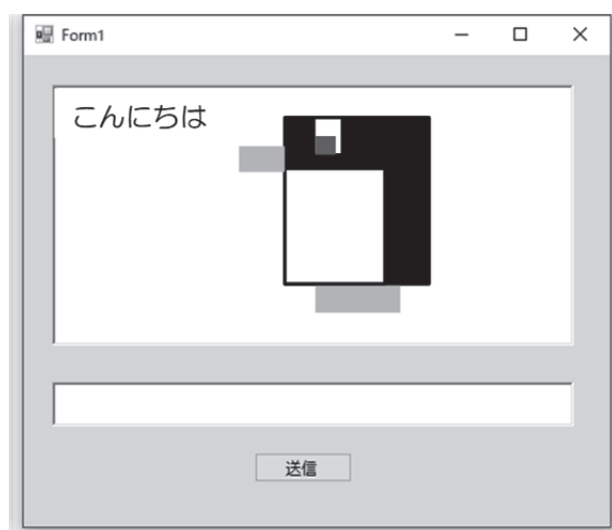


図 2 チェットプログラムの実行例

- .NET Framework
- 変数型（整数型、実数型、論理値型、文字型、文字列型、それぞれの変換）
- 配列
- 演算子
- 関数
- 乱数

2.2 無線通信プログラム ジグザグ (Zig) に飛ぶミツバチ (Bee) を形容する ZigBee は、ワイヤレスセンサネットワークの構築に適した低コスト、低消費電力の無線通信規格である。1 つのネットワークに最大 65,535 ノード（端末）が接続できることから、IoT インフラを担うキーテクノロジーとして期待されている。第 2 週では、この ZigBee を用いてチャットプログラムの作成に取り組む。

学生は 2 人一組となり、ZigBee モジュールを各自のノート PC に装着後、メーカーが提供する GUI ツールを利用して P2P 接続する。つぎに Visual C# を用いた「日本語チャット」プログラミングでは、指導書を参照することでシリアルポートの接続、送受信文字用テキストボックスの生成、非同期受信処理の設定を行う。また、オプション機能の「お絵描きソフト」については、描画用キャンバスを生成後、絵のユニットとなる矩形体の色、座標、サイズを設定する仕様とした。各プログラムの動作を確認した後、「文字と絵を送受信できるチャットプログラム」の作成を成績評価の課題として提示している（図 2）。評価項目は実行結果とした。学習項目は下記のとおりである。

- ZigBee の機能と特性



図 3 自動演奏ピアノプログラムの実行例

2.3 シリアル通信プログラム 第3週の実験実習項目は「シリアル通信プログラミング」である。シリアル通信では一本の回線を用いてデータが 1 ビットずつ順番に伝送される。そのためクロックずれが発生しにくくノイズに強い。また、低コストで配線が容易なため、長距離通信にも利用可能である。メカトロニクス分野では、PC、マイコン、センサ等の周辺機器の接続に使用されるケースが多い。

指導書に従い、学生はまず受信文字列（周波数と鳴動時間）に応じて Arduino に装着した他励式圧電ブザーを鳴らすプログラムを作成する。シリアルモニターで動作を確認後、Visual C# で作成したピアノの鍵盤に対してクリックアクションと関数を関連付ける。つぎにそれぞれのプログラムを連動させることで、鍵盤を押下すると意図した音が鳴るよう Arduino を制御する。作成した鍵盤を用いて各学生に任意の曲を演奏させた後、成績評価課題として楽譜ファイル (.csv) を読み込み自動演奏するための通信プログラムを提示している（図 3）。評価項目は実行結果とした。以上、本実験実習の学習項目は下記のとおりである。

- Arduino の機能と特性

2.4 Python プログラミング 第4週では「Python プログラミング」の導入教育を実施している[1]。Linux 環境で動作している Raspberry Pi を用いて[2]、プログラミング画面を共有しながら「階乗計算プログラム」、「平均値計算プログラム」、「円周率計算プログラム」の作成に取り組む。学生全員のプログラムが正しく実行されていることを確認後、成績評価課題として「素数判定プログラム」の作成を提示している。評価項目は実行結果とした。具体的な学習項目は下記のとおりである。

- メタ構文
- Print
- 変数
- 演算子
- 文字列処理
- if 文、while 文、for 文、continue 文、break 文
- 関数

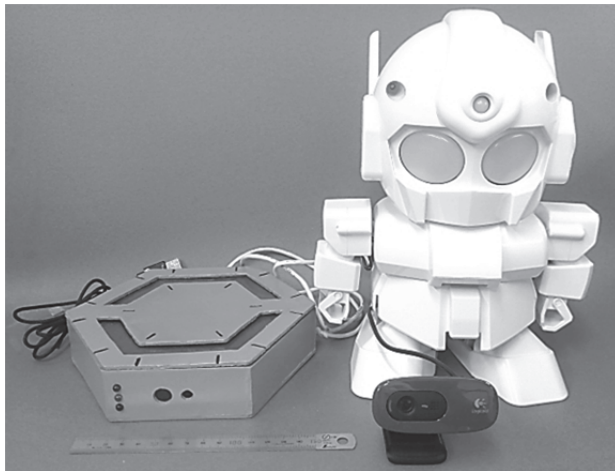


図4 スマートスピーカーと音声制御用ロボット

2.5 音声認識プログラム 第5週の実験実習項目は Raspberry Pi への「音声認識プログラム」の組み込みである[3]。音声認識技術の応用については、Pepper 等のコミュニケーションロボットや Google Home に代表される AI スピーカーがメディアを賑わしているが、今後さらなる展開が期待できる。

解説用の課題はスマートスピーカーの製作とした(図4左)。音声認識エンジン Julius[4]、マイク、スピーカーを設定後、プログラミング画面を参照しながら英語翻訳等の「Q&A プログラム」を Python で作成する。Julius は高性能な音声認識ソフトウェアであり、2 万語彙の読み上げ音声に対して 90%以上の認識率を有している。しかし、デフォルトの辞書では動作が不安定なため、実験専用の音声入力データベースを作成した。プログラムの実行を確認した後、成績評価課題として「ロボット音声制御プログラム」の作成を提示し[5]、両手・両足・首の動作制御の可否を評価項目とした。学習項目は下記のとおりである。

- ・ Raspberry Pi の機能と特性
- ・ Linux OS の操作方法

3. 学生アンケート

次年度以降の改善点を抽出するために、すべての実験実習項目についてアンケートを実施した。授業最終週に Google Form で作成したアンケートフォームを QR コードで配布することにより、学生からの回答はオンラインで回収した。アンケートの内容は以下の 3 項目とした。まず、「プログラミングへの興味とプログラミングスキル」に関しては、実験実習前後での変化を「ある」、「ややある」、「どちらでもない」、「ややない」、「ない」の 5 段階で調査した。つぎに「C#を用いたプログラミング実験実

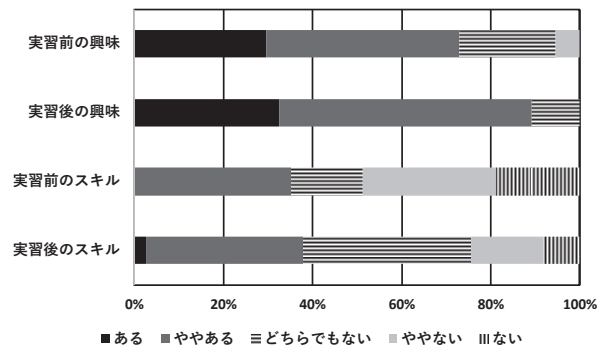


図5 興味とスキルに関するアンケート結果

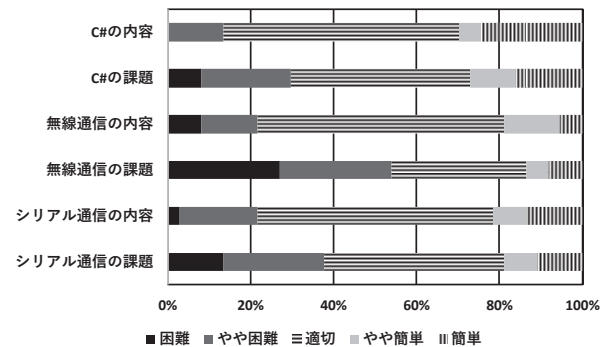


図6 C#プログラミングに関するアンケート結果

習」および「Python を用いたプログラミング実験実習」について、内容と課題の難易度を「困難」、「やや困難」、「どちらでもない」、「やや簡単」、「簡単」の 5 段階で評価した。対象はメカトロニクスコース4年生の39名とし、37名から有効回答を得た。

3.1 プログラミングへの興味 図5に示すように、プログラミングに興味は「ある」、「ややある」と回答した学生は、実験実習前後で全体の70%から90%に増加した。この結果は、現在の実験実習内容がプログラミングに消極的な学生の動機付けになることを示唆している。

3.2 プログラミングスキル 実験実習前、プログラミングスキルが「ない」、「あまりない」と回答した学生は全体の約50%を占めていたが、実験実習後にはその数が25%まで低下した(図5)。一方、「どちらでもない」を選択した学生が大幅に増加したのに対して、「ややある」、「ある」の合計数にほぼ変化はなかった。その理由として以下の2点が考えられる。

(1)本実験実習は基本的なプログラミング技術の向上を目的としているため、スキルがあると自覚している学生が実験実習前後で明確に「これができるようになった」とは考えにくい。

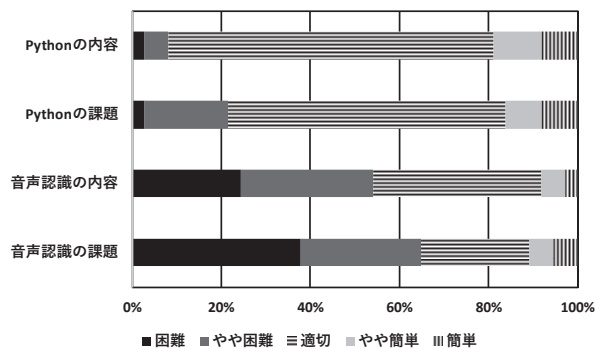


図 7 Python プログラミングに関するアンケート結果

(2)高難度の課題についてはクリアできる学生が少数であったため、「〇〇〇のスキルが向上した」と実感できない。

3.3 C#プログラミング 図 6 に示すように、内容については「簡単」、「やや簡単」、「適切」と回答した学生が全体の約 90%に達しており、解説を聞きながら同時進行で作成する「カプセルトイシミュレーター」の難易度設定は適切であったと結論できる。演習課題である「ブラックジャックシミュレーター」の作成に関しては「困難」、「やや困難」を選択した学生が増加したが、教育的効果という観点からは妥当な結果と考える。

3.4 無線通信プログラム 内容については約 80%の学生が「簡単」、「やや簡単」、「適切」と回答しており(図 6)、「日本語チャット」および「お絵描きソフト」の指導書が適切に作成されていることを示唆している。一方、「日本語チャット」と「お絵描きソフト」を統合する課題については「困難」、「やや困難」を選択した学生が 50%を超えた。この結果については、(1)コード数および似たコードが多いため入力ミスが頻発する、(2)通信できない原因が ZigBee 設定とプログラムのいずれにあるのかを判断しづらい等の理由が考えられる。

3.5 シリアル通信プログラム 内容については「簡単」、「やや簡単」、「適切」と回答した学生が全体の約 80%に達した(図 6)。このことから、指導書を参照して取り組むピアノの「手動演奏プログラム」の難易度は適切に設定されていると結論できる。一方、「自動演奏」の課題に関しては「困難」、「やや困難」を選択した学生が 4 割を超えた。手動演奏についてはすべての学生が容易に達成したことから、実際の楽譜を参照しながら csv ファイルを作成する作業に煩わしさを感じたものと考ええる。

3.6 Python プログラミング 図 7 に示すように、内容については 90%以上の学生が「簡単」、「やや簡単」、「適切」と回答した。ほぼ全員が初学者であったことを考えると、C 言語の基礎知識があれば Python の導入は容易であると結論できる。課題である「素数判定プログラム」の作成に関しては「やや困難」と感じる学生が増加したが、導入教育の直後であることから妥当な結果といえる。

3.7 音声認識プログラム 内容については「困難」、「やや困難」と回答した学生が全体の 50%を超えた(図 7)。課題に関しても 6 割を超える学生が同様の感想を有していた。その理由として、(1)Raspberry Pi を SSH 接続で制御する際に使用する Linux の CUI に慣れていない、(2)長いコード(約 100 行)を入力しなければならない、(3)バックグラウンドで実行するサービスが正常に動作しない等が考えられる。

4. まとめ

本稿では、メカトロニクスコースの 4 年次「電子機械工学実験 I」で実施しているプログラミング実験実習の内容を紹介した。C#プログラミング(無線通信とシリアル通信)と Python プログラミング(音声認識)はそれぞれ 5 年目、1 年目の取り組みであり、「さまざまな応用分野の提示」および「苦手意識を持つ学生のボトムアップ」を目的に展開している。アンケートの結果から、おおむねその目的は達成できたものと考えているが、その一方で、「習得スキル」を明確にできていない点は改善すべきである。次年度以降、学生自身が「今日の実験実習で〇〇〇ができるようになった」と実感できるよう、課題の内容あるいは難易度を見直したい。

参考文献

- [1] “Welcome to Python.org” <https://www.python.org/>
- [2] “Teach, Learn, and Make with Raspberry Pi” <https://www.raspberrypi.org/>
- [3] 「ラズパイに、人の声を理解させる。」 <http://www.asahi-net.or.jp/~cb9i-kn/geek/20170513toext.html>
- [4] 「大語彙連続音声認識エンジン Julius」 <http://julius.osdn.jp/>
- [5] “RAPIRO wiki” <http://wiki.rapiro.com/page/index/>