

プログラムによる計測・制御のための Raspberry Pi 用写経型学習教材の開発

辻仁志^{†1} 喜多一^{†2}

日本の新学習指導要領では、中学校の教科「技術・家庭」の技術分野で、「プログラムによる計測・制御」が必修化されるなど、プログラミング教育は重要な研究対象であるが、初学者には様々な躓きがあり多くの支援が必要である。教育の負担を減らしつつ、学習効果を高めるためには、生徒が自立的にプログラミングを学習でき、プログラミングを行える能力を習得することが望ましい。本研究では、Raspberry Pi を活用する形で、「プログラムによる計測・制御」の学習を目的に、写経型学習という手法を用いたプログラミング教育の教材の開発を行う。この手法を用いた教材を利用することで、初学者の学習効率を向上させ、教員がプログラミングを教える難しさを軽減できると考えられる。作成した教材を普通科高等学校での課外活動で試用することで評価を行う。

Development of Shakyo-style Learning Material for Measurement and Control by Computer Program using Raspberry Pi

HITOSHI TSUJI^{†1} HAJIME KITA^{†2}

Education of computer programming is an important research issue in general education as well as vocational education. For example, in Japanese new course of study for junior high-schools, instrumentation and control using computer programming has been changed from an optional unit to a compulsory one in the subject “Technology and Home Economics.” However, novices of computer programming face various difficulties in their learning, and effective and efficient teaching must be pursued by making learning of computer programming independent as much as possible. In this paper, the authors have developed learning materials for “instrumentation and control using computer programming” using Raspberry Pi. For development of the material, “Shakyo-style” learning is adopted so as to raise efficiency of learning by novices. The developed material is used and evaluated in extracurricular activities in a senior high-school.

1. はじめに

現在、日本ではIT人材の不足が深刻である。独立行政法人情報処理推進機構が2013年に行った調査[1]では、IT企業のIT人材の「量」について「大幅に不足している」または「やや不足している」と回答した企業が82.2%であった。2009年の49%から急増している。このようにIT人材が不足している現状のなかで、2013年6月に発表した安倍政権の成長戦略[2]では、「4. 世界最高水準のIT社会の実現」内の「⑥産業競争力の源泉となるハイレベルなIT人材の育成・確保」内の1つの項目である「ITを活用した21世紀型スキルの修得」のなかで、「義務教育段階からのプログラミング教育等のIT教育を推進する。」と掲げている。また、新学習指導要領[3]では、中学校の教科「技術・家庭」の技術分野で、「プログラムによる計測・制御」が必修化された。これらを実質化し日本の教育現場でプログラミング教育を推進するには、どのような教育方法が良いのかが議論の対象となる。

初学者へのプログラミング教育は難しく、この分野ではこれまでに多くの研究が行われている[4]が標準的な指針が示されるまでには至っていない。初学者には様々な躓き

があり多くの支援が必要である。さらに、日本の中学校や高等学校では、1人の教員が30人~40人程度の生徒に教育をすることになる。このような現状から、教員による支援の必要頻度をなるべく少なくし、教育の負担を減らしつつ、学習効果を高めたい。そのためには生徒が自立的にプログラミングを学習でき、プログラミング能力を習得することが望ましい。これについて、喜多、岡本らは「写経型学習」[6],[11]という手法を用いたプログラミング教育を提唱している。本研究ではこの手法に着目し、プログラミング教材の開発を行う。これにより初学者の学習効率を向上させ、教員がプログラミングを教える難しさを軽減できると考えられる。

また、本研究ではプログラミング教育に2012年に発売された名刺サイズのPCボード「Raspberry Pi」を活用する。Raspberry Piは発売以来、世界で300万台以上が出荷されている。日本では、2013年10月に、GoogleがNPO法人CANVASと協力し、Raspberry Piを活用した「コンピューターに親しもう」プログラム（現名称：PEG）を開始することを発表したことで、話題になった。Raspberry Piの強みは、単独で動作する極めて安価なパソコンであることや、GPIO（汎用入出力）が使えるため様々なセンサーやアクチュエータと接続できること、パソコン・電子工作・サーバなど多様な使い方ができること等が挙げられる。上記の強みから、中学校での「技術・家庭」技術分野での「プログ

^{†1} 京都大学大学院 情報学研究科
Graduate School of Informatics, Kyoto University

^{†2} 京都大学 国際高等教育院
Institute for Liberal Arts and Sciences, Kyoto University

ラムによる計測・制御」や、高等学校での教科「情報」などに Raspberry Pi を活用することが期待できる。さらには、GPIO を活用することで中学校や高等学校での理科教育にも活用できる可能性があり、今後は多くの学校で活用される可能性がある。

2. 研究の目的

本研究では、写経型学習教材[6]の指導方法に従い、Raspberry Pi で利用できる、初学者向けプログラミング教材の開発と評価を行う。プログラミング言語には、Raspberry Pi の開発者が推奨している Python を想定した。また、開発する教材は Raspberry Pi に対応したもので、GPIO を活用した組み込みプログラミングを含めた。作成した教材の評価は普通科高等学校での課外活動で試用実践を通じて行った。

3. 写経型学習

写経型学習とは、多くのサンプルプログラムをすべて入力し、実行することを求めた上で、その後は課題を与えてプログラムを構成させる学習手法である。岡本ら[5]は、プログラミング教育に課題達成型の学習を取り入れた場合、プログラミング言語自体の学習は、学習者による自習にゆだねることになり、学習者がプログラミング言語を自習することを想定した教育支援を検討することが大変重要になってくると述べている。そして初学者が指導者による一定の指導は受けられることは想定しつつ、自習を中心にプログラミングを習得できるような教材を開発した[5]。また、岡本ら[9]は、プログラミングの「写経型学習」における初学者の躓きを次のように類型化している。すなわち写経型学習を遂行する上で自立的に作業することができない「作業の自立性」に係る躓きと、写経型学習の過程から目的とする内容を学び取ることができない「作業を介した理解に係る躓き」の2つに類型化し、さらに、本質的な認知的負荷を伴う学習過程による躓きと、非本質的な認知的負荷を伴う学習過程による躓きの2つに区別している。これらの躓きを排除または軽減するために、テキストの開発においては自立的な作業遂行のための、「コンパイルの手順やエラー発見の方法についてその手順を掲載すること」と、作業介した理解のために「学習の目的、学習の仕方など、学ぶ内容とともに学び方を明示すること」という2つの方略がなされている。

上記の研究で開発した教材をもとに、喜多・岡本らが開発したものが、「写経型学習による C 言語ワークブック」[6]である。喜多ら[7],[11]は、この教材の指導方針として、以下の項目をあげている。

- プログラミングと用いられる記号や予約語の音読、if 文や for 文などの日本語での訳読を通じた、記号・略号を多用するプログラミングへの慣れの促進と教授者とのコミュニケーションの改善。

- 学習者が自習により、プログラムの入力・実行することで、プログラムに触れる経験を高めるための多数の実行可能なソースコードと実行結果の教材としての提示。
- プログラミングの自学自習での躓きを低減するための文法誤り例の積極的な経験
- ソースコードの逐次解説と実用に重点を置き、図解を多用した文法的事項の解説。

さらに自学自習をするプログラミング言語の文法的事項の学習を行ったのち、

- プログラムによる問題解決過程を具体例に沿ってなぞりながら、プログラムを全体として構成するプロセスを経験する。

という構成をとっている。

喜多らはプログラミング教育の手法として開発した写経型学習をフィジカルコンピューティングの科目に取り入れた[7]。この科目は、京都大学のポケットゼミ「Physical Computing 入門」として2011年に開講されたものである。この科目は、前半でワンボードマイコン Arduino とブレッドボードを用いた電子回路の製作とプログラミングについて基礎的事項を学習し、後半では使用する部品の予算制約の範囲内で自由な発想で作品を構成し発表する。また、この授業案は3日間での集中実施する講座にも適用可能だと述べている。

4. Raspberry Pi

Raspberry Pi (図1) はもともと Eben Upton ら[8]がケンブリッジ大学のコンピュータサイエンスの履修課程の志願者向けに、安くて簡単にプログラミングをできる環境として開発したものである。Raspberry Pi の開発者 Upton は、センサー、モーター、ライト、マイクロコントローラなどを使ってシステムを組み立てる物理的なコンピューティングをとつともなくおもしろいことだと考えており、そのため GPIO が付いている。別途マイコンボードを購入しなくても、すぐに物理的なコンピューティングを始めることができる。

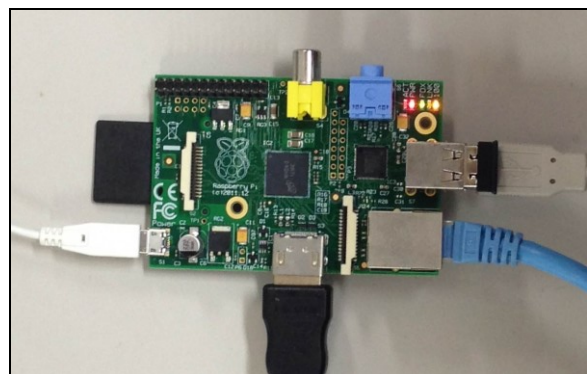


図1 Raspberry Pi

現在, Raspberry Pi は世界中で流行しており, プログラミング教育だけに留まらず, 様々な活用法が開発されている. 例えば, Raspberry Pi 用カメラモジュールを利用して, 監視カメラを作成したり, 温度センサーなどの各種センサーと接続して各情報を記録したりできる. また, Raspberry Pi 開発者 Upton はホームシアターPC としての活用も一般的としている[8]. 日本のプログラミング教育の現場では, 2013年10月に, Google が NPO 法人 CANVAS と協力し, Raspberry Pi を活用した「コンピューターに親しもう」プログラム(現名称: PEG)を開始した[9]. プログラミング学習プロジェクト PEG (Programming education gathering) では, 多くの小学校, 中学校, 高等学校と連携し, Raspberry Pi を授業に活用している. また, 大学では筑波大学が「分野・地域を越えた実践的情報教育協働ネットワーク」(通称 enPiT)の事業内の科目の「組込みプログラム開発」で Raspberry Pi が活用されている実践例がある[12].

5. 開発した教材及びカリキュラムのデザイン

5.1 開発した教材

開発した教材は, C 言語の写経型学習教材[6]を参考にし, Raspberry Pi, Python 向けに内容を調整する形で開発した(図2,図3). python のプログラミングには OS 「Raspbian」に標準でインストールされている「Python IDLE3」を利用する. 第3章の後半では, よくあるエラーの実例とエラーメッセージを例示した(図4).

また, 練習問題として, リストを使わずに記述したサンプルプログラムをリスト使って短いプログラムを作成させる練習問題や, for 文とリストを使わずに記述したサンプルプログラムを, for 文とリストを使って短いプログラム改変させる問題を導入している. さらに13章に, 四角形や円などの図形を描く方法などのサンプルプログラムと練習問題を追加した.

さらに, 学習項目特有で躓きやすいポイント (if 文での「=」と「==」の間違い, for 文でのインデントのずれなど)には, エラーの例を各学習項目において, 例示するようにしている. 「=」と「==」の間違いを経験させるサンプルプログラムを図5示す. このようにサンプルプログラムにエラーのプログラムを含ませることで, 学習者は必ずエラーを明示的に経験することができ, 自習時のエラーによる躓きを低減させることができる.

教材は, 開発中であり, 現時点で17章まで作成している. 開発した教材の目次とサンプルプログラム数を表1に示す.

5.2 カリキュラム

学習内容は初学者がプログラミングの基本を学ぶカリキュラムとして構成した. 初学者を対象としているため, 文字列操作, ファイル操作, クラスとオブジェクトなどの項

第8章 条件によって処理を分ける方法 (if 文)

8章の学習目的
 1. 条件によって処理を分ける方法 (if 文)を理解する.

プログラムリスト(1) ファイル名: sample8_1.py
 ・年齢をキーボードから入力して, 飲酒できる年齢かどうかを判断します.
 ・6行目と8行目は「TAB (タブ)」キーを1回押してから print 関数を書いてください.
 ただし, 本書のように Python IDLE 等のエディタを使用している場合は, 5行目入力後の Enter を押すと, 6行目の先頭に自動的に「TAB」を押した時と同じ空白が挿入されますので, 「TAB」を押す必要はありません. 7行目の「else」は左端から記述してください.

行数	命令	解説
1	age = input('あなたの年齢は?: ')	
2	age = int(age)	
3	print ('あなたの年齢は{0:d}歳です'.format(age))	
4		
5	if age >= 20:	age が 20 以上の場合
6	print ('飲酒できます。')	「飲酒できます」と表示
7	else:	age が 20 以上ではない場合
8	print ('未成年なので飲酒できません。')	「未成年なので飲酒できません」と表示

チェック

実行結果(1.1)

```
あなたの年齢は?: 20
あなたの年齢は 20 歳です
飲酒できます。
```

チェック

実行結果(1.2)

```
あなたの年齢は?: 16
あなたの年齢は 16 歳です。
未成年なので飲酒できません。
```

チェック

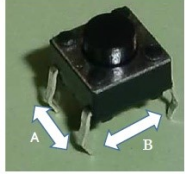
1

図2 開発した教材

第15章 タクトスイッチとLEDを使った回路の制御

15章の学習目的
 1. ブレッドボード上で LED とタクトスイッチを使った回路を作成し Raspberry Pi の GPIO で制御する。(デジタルの入力と出力)。

タクトスイッチ
 今回の実習ではタクトスイッチ(右の写真)の押し下げによって LED の点灯, 消灯を制御する回路を作ってみます. タクトスイッチには4本の足が出ていますが, B 方向の2本は内部で接続されています. スイッチを押し下げると A 方向の2本が導通します.



ブレッドボード上での配線
 Raspberry Pi でタクトスイッチの押し下げ状態を検出するために図1のような回路を, ブレッドボードを使って作成します. 以下の手順に従ってください.

注意! 回路を構成している時は Raspberry Pi をブレッドボードと接続しているケーブルから外しておいてください. これは電源などを短絡して Raspberry Pi や電子部品を壊さないようにするためです.

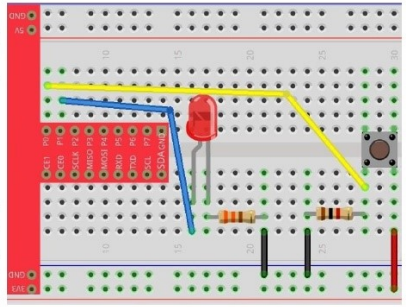


図1 タクトスイッチとLEDの配線図

注意! 電子部品の端子は柔らかくて曲がり折れたりしやすいものです. ブレッドボードに差し込む際には無理に押し込んだりせず, 慎重に作業してください.

図3 開発した教材

電子回路を含むプログラミングを行う第 14 章以降は、2 人でペアを組む形で進める。これは、電子回路の配線とプログラムのソースコードを相互に確認してもらうためである。特に電子回路の配線ミスは、電源の短絡、部品への過電圧などで、本体や部品の破損を招くことから、ペアでの確認作業は非常に重要である。単に誤りを低減するというだけでなく、単独での学習が中心の高校生に対してペアでの作業による相互チェックという方法そのものを理解してもらう意図も含まれている。

表 1 開発した教材

章	目次	例題数
1 章	学習の仕方を理解する	-
2 章	プログラミングに使用する文字と用語	-
3 章	Raspberry Pi の初期設定とプログラムの開発手順	1
4 章	画面に表示する方法	11
5 章	データを入力する方法	4
6 章	値を保持する方法(変数)	8
7 章	処理の流れとフローチャートを理解する	-
8 章	条件によって処理を分ける方法(if 文)	5
9 章	繰り返し処理 (for 文) と変数を一括管理する方法 (リスト)	22
10 章	リストの便利な使い方	12
11 章	while 文	2
12 章	関数	9
13 章	図形を描く方法	1
14 章	電子回路の作成への準備	-
15 章	タクトスイッチと LED を使った回路の制御	4
16 章	圧電ブザーを使って音を鳴らす	1
17 章	アナログ出力・アナログ入力	2

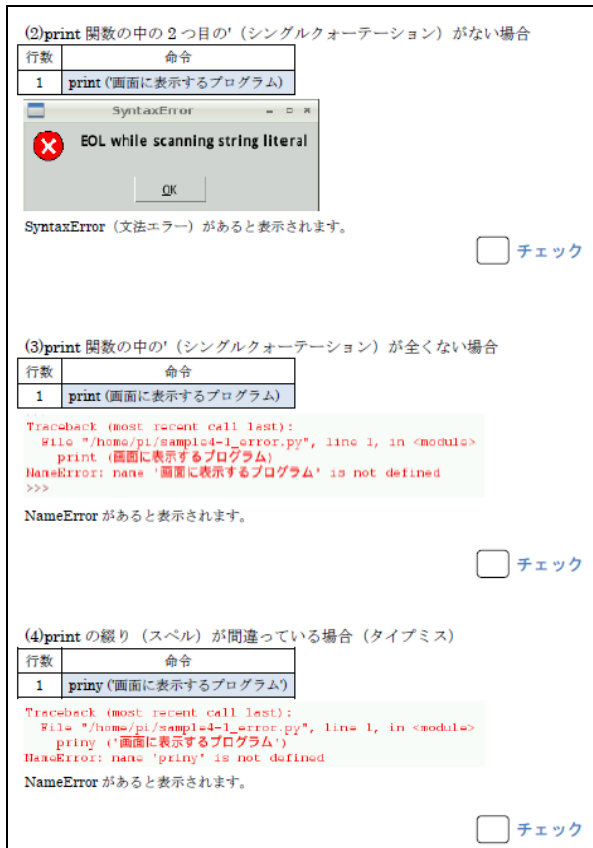


図 4 よくあるエラーの実例とエラーメッセージ

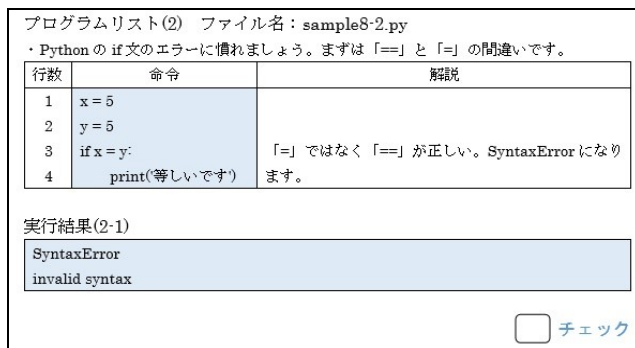


図 5 if 文でよくある間違い

目については扱っていない。Python のプログラミングを学習する第 4 章から第 13 章までは、タイピングスピードや理解する速さに違いがあるため進捗は生徒ごとに違う。教材そのものは極力、学習者のペースに合わせて学習できるように構成しているが、実践では Raspberry Pi が利用できるのは教室内に限られるため当該の回に配布した資料に沿って学習を終了できないこともある。その場合は、大幅に進捗が遅れない限り次の週に続きを行っても構わないことにした。また、前回欠席者には、サンプルプログラムを全て実行してもらうのではなく、著者が適宜、最低限実行してほしいプログラムを指示して、学習してもらうように調整した。

6. 高等学校でのプログラミング講習の実践

6.1 講習の概要

大阪府立天王寺高等学校は、放課後の時間に 1 回に 1 時間 30 分～2 時間のプログラミングの講習会を実施している。天王寺高等学校情報研究部の顧問の教諭に協力して頂き、開発した教材を試用した。また、教室は天王寺高等学校の情報教室を利用している。同教諭が立ち会いのもと、著者が TA (ティーチングアシスタント) として生徒達を支援している。2014 年 6 月から、月に 2～3 回程度開催し、1 回目は Raspberry Pi への OS のインストールや設定などを行い、7 月までにプログラミングの学習を実施し、9 月と 10 月で GPIO を活用した電子回路を含むプログラミングを実施した。最終的には、生徒達のアイデアで作品を構成してもらう予定となっている。現時点までに実施した講習会の日付と配布した教材と参加人数を表 2 で示す。

表2 実施した講習会で配布した教材と参加人数

講習会	日付	使用した教材	人数
第1回	6/3	1章～3章(前半)	15
第2回	6/18	3章(後半)～5章	16
第3回	7/2	6章～8章	11
第4回	7/9	9章	9
第5回	7/16	10章	10
第6回	7/28	11章～13章	6
第7回	9/10	14章, 15章	10
第8回	9/17	16章	9
第9回	10/15	17章	3

6.2 データの収集

生徒達は、著者が開発した教材を見ながら、自学自習でプログラミングを学習する。その時、躓きがあれば著者(辻)が支援を行い、その内容を記録している。また、毎回の講習終了時には、アンケートを回収している。

6.3 受講生の構成

本講習の受講生は、17名である。放課後に実施しているため、体育系の部活動に参加している生徒が練習などと重なることなどがあり、全員が毎回参加できるわけではない。初回に事前アンケートを回収した。受講生の内訳は、1年生7名、2年生10名であった。また、プログラミング経験についてのアンケート結果を表3に示す。受講生のなかには、情報研究部に所属している生徒が含まれていることと、中学校の技術・家庭の授業で学習したことがある人がいるため、プログラミングの経験者が9名含まれている。また、学んだことがあるプログラム言語についてのアンケート結果を表4に示す。今回の講習会で利用するPythonの経験者は0であった。第14章以降では、電子回路を扱うので、電子工作の経験についてのアンケートも実施した。アンケート結果を表5に示す。

表3 プログラミング経験の有無(複数選択可)

回答者数 17

回答項目	人数
中学校の技術の授業で学んだ	3
中学校の技術以外の授業で学んだ	1
高校の情報の授業で学んだ	1
高校の情報以外の授業で学んだ	0
小学校の授業で学んだ	0
その他教えてもらう機会があった	5
自分勉強した	2
プログラミング経験はない	7
未回答	1

表4 学んだことがあるプログラム言語は何ですか(複数選択可) 回答者数 17

回答項目	人数
C	6
Java	2
C++	1
JavaScript	1
その他	1
C#, Basic, Visual Basic, Ruby, Python, Perl	0

表5 電子工作の経験はありますか(複数選択可)

回答者数 17

回答項目	人数
中学校の技術の授業で、回路図を見て指示通りに作成した	9
中学校の技術の授業で、授業で回路図を作成して、作品を作った	1
中学校の技術以外の授業で、回路図を見て指示通りに作成した	1
中学校の技術以外の授業で、授業で回路図を作成して、作品を作った	0
高校の授業で学んだ	0
小学校の授業で学んだ	0
その他教えてもらう機会があった	3
家などで、自ら電子工作に取り組んだ	1
経験はない	4

7. プログラミング講習の実践結果

7.1 講習会各回の参加人数

講習会には、情報研究部に所属している生徒も参加しているが、既にプログラミングができる人がいる。このような生徒は出席したが、本研究で開発した教材のサンプルプログラムを実行していない人もいるので、サンプルプログラムを実行し、受講後アンケートを提出した人のみを参加者とする。このように参加人数にばらつきがあるのは、前述のとおり講習会を放課後に実施しているため、体育系の部活動に参加している生徒が練習などと重なることや、他の講習や、学校行事で一部の生徒が受講できなかったことによる。

7.2 学習の進捗状況と躓きの抽出

以下ではデータ整理が終了した第1回～第7回の講習について実施状況とそこで見られた学習者の躓きを述べる。

7.2.1 講習会第1回

講習会第1回では、Raspberry PiへのOS「Raspbian」のインストールと日本語入力のための設定など、基本的な設

定を自立的に行えるように、教材を作成し配布した。受講者には教材の指示通りに OS のインストールと設定を行うように指示した。プログラミングは講習会第 2 回からであり、そのため第 1 回は受講後アンケートを実施していない。

7.2.2 講習会第 2 回

講習会第 2 回では、Python IDLE3 の開始方法、エディタの開き方、ファイルの保存方法、プログラムの実行方法を指示した教材を確認した後、受講生はサンプルプログラムを打ち込み、自分のペースで学習してもらった。4 章の最後まで進めた人は 16 名中 14 名、5 章の最後まで進めた人は 16 名中 6 名であった。正確に計測・比較はしていないが大学での学部学生のコンピュータを用いた演習などと比べ、タイピング速度が遅いように感じた。理解度を問うアンケートでは、理解できた (6 名)、ほとんど理解できた (10 名)、半分ほど理解できた (0 名)、ほとんど理解できなかった (0 名)、全く理解できなかった (0 名) であった。

著者 (辻) が支援した内容として、(1)プログラムのエラーが解決できない。(2)実行の仕方がわからない。(3)動作トラブル。(4)テキストが理解できない。の大きく分けて 4 つがあった。(1)では、`print` 関数の「(」が全角になっていたり、文字列を囲む「'」(シングルクォーテーション)が全角になっていたり、「.」(ドット)の入力し忘れなどがあった。(2)では、`input` 関数を利用するときに、入力する場所がわからないといった問い合わせがあった。(3)では、急に文字が入力できなくなるといった動作トラブルがあった。(4)では、書式指定をする意味がわからないという指摘があった。

自由記述アンケート「理解できなかったところがあれば教えてください。」では、「`int` と `float` の違いがわからない」という回答が 1 名からあった。「理解するのに、時間がかかったところがあれば教えてください。」では、「書式指定」が 3 名、「`input()`」、「関数の意味」、「文字列と変数の違い」という回答が見られた。「エラーが起きたところがあれば、どのプログラムのどのようなミスでエラーが起きたか教えてください。」では、「クォーテーションを全角にしていた」、「「{」が「(」になっていた」、「カンマとドットの区別がつきにくく `TYPE` エラーが出続けていた」、「「」2 つを「"」にしていた」、「全角・半角のミス」などの回答が見られた。

感想を聞いた自由記述には、「自分で入力したものが、きちんと意図したものになって出力されるのがとてもうれしくて興奮しました。」、「細かいところまで書いていて分かりやすかったから、おもしろかった」、「解説に書いていることが現実になるのが楽しい」、「気になった事を調べられてよかった。初めてプログラムして、おもしろかった。」、「意図したことがちゃんとできたときは嬉しかった。もっと勉強したいです。」、「あまり難しくなく、楽しく進められた」、「初めてのプログラミングでとても興味深かった。」

「ずっとプログラミングをやってみたかったので、入力していることじたいが楽しかった」、「自分が入力したモノがどんどん画面に表示されていくのがとてもおもしろかった」、「ただの文字が動きをとまってシナリオ通りに動くのが嬉しかった」、などの感想が見られた。

「ずっとプログラミングをやってみたかったので、入力していることじたいが楽しかった」、「自分が入力したモノがどんどん画面に表示されていくのがとてもおもしろかった」、「ただの文字が動きをとまってシナリオ通りに動くのが嬉しかった」、などの感想が見られた。

7.2.3 講習会第 3 回

講習会第 3 回では、前回の続きから受講生はサンプルプログラムを打ち込み、自分のペースで学習してもらった。6 章の最後まで進めた人は 11 名中 4 名、8 章の最後まで進めた人は 11 名中 2 名であった。理解度を問うアンケートでは、理解できた (5 名)、ほとんど理解できた (5 名)、半分ほど理解できた (1 名)、ほとんど理解できなかった (0 名)、全く理解できなかった (0 名) であった。

著者 (辻) が支援した内容として、(1)プログラムのエラーが解決できない。(2)テキストが理解できない。の大きく分けて 2 つがあった。(1)では、変数名のスペルミスが多かった。また、「=」と入力するところで「-」と入力していたエラーもあった。(2)については、変数について理解が不足している人がいた。

自由記述アンケート「理解できなかったところがあれば教えてください。」では、「`sample6-5` と `6-6` が答えが同じなのに、命令文が違うのがいまいち理解できなかった (注: `sample6-6` は `sample6-5` を少ない変数で構成したプログラム)」という回答が 1 名から得られた。「理解するのに、時間がかかったところがあれば教えてください。」では、「書式指定の「`5.1f`」」、「`=`」や「`==`」や「`>=`」がでてきて混乱した」、「`sum = sum + price` で (左辺と右辺で) 同じ変数を使っているところ」などの回答が見られた。「エラーが起きたところがあれば、どのプログラムのどのようなミスでエラーが起きたか教えてください。」では、「スペルミス」、「`if` の後ろにスペースがなかった」などの回答が見られた。

感想を聞いた自由記述には、「8 章で `if` がつかえてうれしかったです。後から入力した数値でプログラムが判断して出力してくれるのが楽しかったです。」、「長いプログラムを入力したのが楽しかった」、「`#`を使って後で見返しやすくて便利だと思った」、「どんどん色々なことができていってよかったです。」、「長いプログラムが多くて疲れた」、「だんだんタイプするのが疲れてきた」、「少し `python` のしくみがわかってきてうれしい」、「`if` で「はい」と「いいえ」の選択ができて、とても良かったと思います」、「1 から自分で作れて、満足です」などの感想が見られた。

7.2.4 講習会第 4 回

講習会第 4 回では、8 章の最後まで進めた人は 9 名中 8 名、9 章の最後まで進めた人は 9 名中 1 名であった。8 章の最後には、プログラムを作成する練習問題 (入力した個数に応じて割引率を変え、合計金額を計算するプログラム) を用意した。理解度を問うアンケートでは、理解できた (3

名), ほとんど理解できた (6 名), 半分ほど理解できた (0 名), ほとんど理解できなかった (0 名), 全く理解できなかった (0 名) であった。

著者 (辻) が支援した内容として, (1)プログラムのエラーが解決できない. (2)練習問題の割引の計算式を自力で作れない. の大きく分けて 2 つがあった. (1)では, if の直後の半角スペースの入力忘れ, スペースが全角になっていた, 「」を全角にしていたなどがあった. (2)は, if 文の使い方は理解できているものの, 合計金額を求める計算式が作れないといったものである.

自由記述アンケート「理解できなかったところがあれば教えてください。」では, 回答者はいなかった. 「理解するのに, 時間がかかったところがあれば教えてください。」では, 「range()関数」, 「変数の置き方」, 「練習問題の割引の計算」という回答が見られた. また, 「エラーが起きたところがあれば, どのプログラムのどのようなミスでエラーが起きたか教えてください。」では, 「if を複数回使ってしまった. (注: 本来は `elseif` を使うべき場所で)」, 「if 文のコロンの場所のミス」などの回答が見られた.

感想を聞いた自由記述には, 「テキストの文字をうつすのではなくて, 自分で考えてプログラムするのが, 難しかったけど楽しかったです. これからもっと使いこなせるようになりたいです」, 「8-6 (練習問題) が楽しかった. 全部自分ではできなかったけれど, 動いたとき感動した」, 「プログラミングが長くなればなるほどミスが多くなったので, 次からは気をつけたい」, 「繰り返す回数を巨大にして, 流れていくのを見るのが面白かった. (ループ回数をとても大きな数にしていた)」, 「難しい計算があって, とてもやりがいを感じることができました。」などの感想が見られた.

7.2.5 講習会第 5 回

講習会第 5 回では, 9 章の最後まで進めた人は 10 名中 7 名, 10 章の最後まで進めた人は 10 名中 4 名であった. 10 章の最後には, リストを使わずに記述したサンプルプログラムをリスト使って短いプログラムを作成させる練習問題と, for 文とリストを使わずに記述したサンプルプログラムを, for 文とリストを使って短いプログラム作成させる練習問題を用意した. 理解度を問うアンケートでは, 理解できた (3 名), ほとんど理解できた (6 名), 半分ほど理解できた (1 名), ほとんど理解できなかった (0 名), 全く理解できなかった (0 名) であった.

著者 (辻) が支援した内容として, (1)プログラムのエラーが解決できない. (2)想定している出力にならない. の大きく分けて 2 つがあった. (1)では, 書式指定に関するものと, for 文とリストを使う練習問題で, Python 特有のリストを順に取り出す for 文と, カウンタ変数 `i` を 0 から順に増やしていく for 文の違いを理解していない躓きがみられた. (2)は, if 文を使うべきところで, 使っていなかったことで発生した.

自由記述アンケート「理解できなかったところがあれば教えてください。」では, 「リストの足し算のプログラム」という回答が 1 名から得られた. 「理解するのに, 時間がかかったところがあれば教えてください。」では, 「for 文」が 2 名, 「for 文の中盤以降」, 「リストを使う練習問題」, 「int 関数(6 章)」, 「8 章(if 文)の最後のサンプルプログラム」などの回答が見られた. 「エラーが起きたところがあれば, どのプログラムのどのようなミスでエラーが起きたか教えてください。」では, 「スペルミス」, 「`i` を `bill_list[i]` としていた. `int` を使い忘れていた.」, 「リストの入力忘れ」, 「値を整数に変えていなかったこと」, 「in range の in 抜け」, 「書式指定」, 「大文字・小文字のミス」などの回答が見られた.

感想を聞いた自由記述には, 「for の使い方がちゃんと理解できていなかったのが, プログラムをつくってみてはじめて気付いたのでよかったです.」, 「長かったプログラムがすごく短くなったのが, おもしろかった.」, 「タイピングが早くなってきたけど, まだまだミスが多くてエラーばかりなので, 正確に打ち込めるようになりたいです.」, 「同じ式が続くと, 集中が途切れてしまった. だらだらやってはいけない」, 「8-6 で自分で考えてプログラムをくんだことがおもしろかったです.」, 「とばしたけどたくさん進めてよかったです. まだ覚えきれていないので, 覚えていきたい. (注: 前回欠席者)」, 「目が疲れた」, 「自分のプログラムも作れたので楽しかったです.」などの感想が見られた.

7.2.6 講習会第 6 回

講習会第 6 回では, 喜多も参与観察に入り, 辻とともに支援を行った. 12 章の最後まで進めた人は 6 名中 4 名, 13 章の最後まで進めた人は 6 名中 1 名であった. 13 章の最後には, for 文と if 文を用いて図形を描く練習問題を用意した. 理解度を問うアンケートでは, 理解できた (1 名), ほとんど理解できた (2 名), 半分ほど理解できた (2 名), ほとんど理解できなかった (1 名), 全く理解できなかった (0 名) であった.

著者らが支援した内容として, (1)プログラムのエラーが解決できない. (2)サンプルプログラムの計算式が理解できない. (3)for 文を使うことを想定している練習問題で for 文を使わずにプログラムを作成していた. (4)練習問題で図形を描く位置を決める引数に変数を使うことに思い至らなかった. (5)for 文の中でカウンタ変数をインクリメントしていた. などがあった. (1)では, 書式指定のミス, スペルミス, 「`=`」を「`=`」にするミス, 「」を全角にするミスなどが見られた.

自由記述アンケート「理解できなかったところがあれば教えてください。」では, 「list」, 「list for 文」の回答が見られた. 「理解するのに, 時間がかかったところがあれば教えてください。」では, 「全体的に時間がかかりました」, 「関数」, 「for 文」, 「リスト 図形 for 文」などの回答が見られた.

た。「エラーが起きたところがあれば、どのプログラムのどのようなミスでエラーが起きたか教えてください。」では、「==のところを=にしていた。blue を biue にしていた。」、「半角を全角にしていた。」、「print に=をつけてしまったこと」、「半角全角 .format」などの回答が見られた。

感想を聞いた自由記述には、「13 章の課題が、先生に教えていただきながらも自力でとけたので、とても達成感がありました。」、「sample13 で式の使い方は分かったので良かった。」、「難しく、やりがいがあって、とても勉強になったと思います。」、「計算や表示だけでなく、図形までかけるということにおどろいた。」などの感想が見られた

7.2.7 講習会第 7 回

講習会第 7 回では、前回欠席者や 13 章の最後まで終わらせていない生徒も、第 14 章から取り組んでもらうことにしている。前回欠席者には、第 13 章までを配布し、講習会時間外で目を通すように指示した。講習会開始時に、2 人 1 組のペアを組み、教材を参照しながら GPIO を利用するためのモジュールのインストールを行った。インストールが終わった班から、ブレッドボード、LED、抵抗器など貸し出した。まずは、第 15 章の回路を、配線図、回路図通りに組むように指示し、回路が出来た班から著者が確認を行った。著者が回路を正しいと確認できたら、サンプルプログラムを打ち込むように指示した。参加者は 12 名で 6 班に分けたが、2 名が比較的早い時間に退室したため、2 名は対象外とし参加者は 10 名とした。15 章の最後まで進めた班は 6 班中 5 班であった。理解度を問うアンケートでは、理解できた (1 名)、ほとんど理解できた (6 名)、半分ほど理解できた (3 名)、ほとんど理解できなかった (0 名)、全く理解できなかった (0 名) であった。

著者 (辻) が支援した内容として、GPIO の解放忘れによる警告がでるといったものが 4 回、プログラムのインデントが揃っていないものが 2 回であった。

自由記述アンケート「理解できなかったところがあれば教えてください。」では、「ところどころ忘れていた所があって、意味が理解しにくい所があった」、「GPIO」という回答が見られた。「理解するのに、時間がかかったところがあれば教えてください。」では、「回路」、「変数の種類が一気に増えすぎて追いつけぬ」という回答が見られた。「今回のようなミス (プログラムや電子回路) があったかを教えてください。」では、「電子回路の配線忘れ」、「文法ミス」、「インデントがずれていました」、「スペルミス」、「変数の定義忘れ」、「(エディタに書くべきプログラムを) Python Shell に書いてしまった」などの回答が見られた。

感想を聞いた自由記述には、「ランプが光っておもしろかった」、「この手で回路を組み立てられたということに実感がまだ湧かない。不思議やでえ」、「書いたのが現実に起こるのが楽しかった」、「電子部品が出てきたので、中学校の技術の授業を思い出して少しワクワクした。ブレッドボ

ードは部品を挿していけばいいだけなので楽だなと思った」、「夏休み前にやっていたことを使って実際に電子工作できて楽しかった」、「プログラミングが、実物で目に見える形で実感できるのでより実生活に近づいたと感じた。」、「3 秒間つきなさい」と指示をしたら、ちゃんと 3 秒間灯り続ける LED がかわいかったです」、「光ったのがうれしかった」などの感想が見られた。

8. 考察

以上の実践結果から、教材を普通科高等学校での課外活動で試用したところ、学習者の躓きに対して著者が支援する機会はかなり少なかった。概ね生徒は教材に沿って自習が行え、本研究で開発した教材が、初学者の学習効率を向上させ、教員がプログラミングを教える難しさを軽減できていると考えられる。一方で、大学での学部学生との比較から、タイピングが遅いと思われる生徒もいた。そのためサンプルプログラムを入力するのに、時間がかかってしまい、進捗が遅れがでていた。支援した内容としては、スペルミスが多かったがこれにも、いくつかの種類があり、(1) 組み込み関数の関数名のスペルミス、(2) 変数を定義する段階での変数名のスペルミス、(3) 変数を使う段階での変数名のスペルミス、に分類できるが、(2) の例は、正しい変数名の変数を使っている命令でエラー (その変数は定義されていない) になるため、初学者はスペルミスを発見するのが難しいと考えられる。このような躓きに対して、配慮が十分でなかったと考えられる。また、Python はインデントをソースコードのブロック化に用いるが、インデントがずれているミスで、実行時エラーにならないものは誤りの解決が難しいことが見られた。

また、8 章の練習問題で、if 文は理解できているが、計算式を自力で作れないためプログラムを作成できないといった躓きに対しては、できるだけ自力で計算式を導けるように、指導を試みた。指導内容は過剰でも過少でも効果が下がるため適切性の確保を要する。理解度に応じて、問題解決のためのヒントを掲載した資料を配布するなど工夫の余地があると考えている。

10 章の練習問題では、初学者でも for 文とリストを概ね理解できていることが確認できた。改善点として、リストを順に取り出す for 文と、カウンタ変数を 0 から順に増やしていく for 文の違いを理解しやすいように工夫する必要がある。

理解度を問うアンケートについては、「理解できた」、「ほとんど理解できた」とする回答が多く、理解しやすい教材になっていたと考えられる。

9. おわりに

本研究では、写経型学習教材[6]の指導方法に従い、Raspberry Pi で利用できる、初学者向けプログラミング教材

を開発した。開発した教材を普通科高等学校での課外活動で試用した。その結果、初学者の学習効率を向上させ、教員がプログラミングを教える難しさを軽減できていると考えられるが、教材・教授法の改善点も見つかった。

第4章～第17章において、改善を行っていく。以下の事項が実践を通じて得られた教材の改善を要する事項である。

- 例示するエラーの例を増やすこと。
- 変数のスペルミスは考察で述べた2つのパターンがあることを、理解できるように工夫する。
- 書式指定など初学者が理解しにくいところは、解説やサンプルプログラムを増やす。
- リストを順に取り出すfor文と、カウンタ変数を0から順に増やしていくfor文の違いを理解しやすいように工夫すること。
- できるだけ支援を必要とせず、練習問題を解いてもらうために、解き方のヒントとして、日本語でプログラムを説明する「設計書」を用いたヒントを与える。
- 講習時の指導の標準化を図り、教員・TA向けの指導用資料を作成する。

などに取り組む。

また、改善した教材は、10月中旬以降に実施している、大阪府立天王寺高等学校の文理学科のSSH（スーパー・サイエンス・ハイスクール）での活動でRaspberry Piを活用した課題研究に取り組む生徒を対象に、再度試用・評価を進めている。

謝辞 本研究を進めるにあたってご協力頂いた大阪府立天王寺高等学校の教員、生徒の皆様にご心より感謝申し上げます。

参考文献

- [1] 独立行政法人情報処理推進機構：IT人材白書2014
- [2] 総理官邸：成長戦略素案（2013年6月）
- [3] 文部科学省：新学習指導要領
- [4] A.Pears, S.stephen, L.Malmi, E.Adams, J.Bennedsen, M.Devlin and J.Paterson ; A survey of literature on the teaching of introductory programming, ACM SIGCSE, 2007
- [5] 岡本雅子, 村上正行, 喜多一, 吉川直人：初学者を対象とした自習中心のプログラミング教育の教材開発と評価, 情報処理学会研究報告「情報教育シンポジウム SSS2010 論文集」, pp.87-94, 2012
- [6] 喜多一, 岡本雅子, 吉川直人, 藤岡健史：写経型学習によるC言語プログラミングワークブック, 共立出版, 2012
- [7] 喜多一, 岡本雅子：写経型によるフィジカルコンピューティング講座の構成, 大学ICT推進協議会2012年度年次大会論文集, 2012
- [8] Eben Upton, Gareth Halfacree, 株式会社クイープ訳：Raspberry

Pi ユーザーガイド, インプレスジャパン, 2013

[9] PEG, <http://pegpeg.jp/> (2014/9/11 参照)

[10] 岡本雅子, 喜多一：プログラミングの「写経型学習」における初学者のつまづきの類型化とその考察, パイディア, 滋賀大学教育実践研究指導センター紀要, No.22, pp.49-53, 2014

[11] 岡本雅子：模倣の重要性に着目した初学者向けプログラミング教育の研究, 京都大学博士論文, 2014

[12] enPiT, <http://www.enpit.jp/> (2014/9/11 参照)