

シングルボードコンピュータを用いたプログラミング学習環境の開発

松原 正也^{*1}

プログラミング教育の重要性が叫ばれ、その実施が必要とされるなか、プログラミングを簡易に行う環境、とくに学習者共通のテキスト型プログラミング教育を実施するシステム環境を準備することは現在容易ではない。本研究では、小型シングルボードコンピュータ「Raspberry Pi Zero WH」を用いて、容易かつ安価にプログラミング学習を実施できる環境を開発し、PCのみならず、学習者が既に所有しているスマートホンやタブレット端末など多様な端末で利用可能であることを確認した。

〈キーワード〉シングルボードコンピュータ、プログラミング教育、コンピュータ言語、スマートホン、タブレット端末、パーソナルコンピュータ、オープンソース

1 はじめに

プログラミング教育が学校教育において必修事項となることについては、すでに広く社会に知られることになった。2020年度からは、小学校においてプログラミング教育の全面実施となり、文部科学省からも「小学校プログラミング教育の手引(第二版)」¹⁾が示され、子どもたちにおけるプログラミングの必要性、また、その可能性が示されるとともに、基本的な考えや具体的な手法などが解説されている。なお、小学校を対象としていることから、この手引においては、いわゆるビジュアル型プログラミング言語を用いることを想定した内容となっている。

この手引きにも記載されているが、いうまでもなく「コンピュータは人が命令を与えることにより動作」するものである。その命令の与え方としてテキスト型プログラミングへの移行も重要であり、その存在は無視できない²⁾。昨今、プログラミング教育が注目を浴びることになり、Scratch³⁾を代表とするビジュアル型プログラミング教育を実施するための環境は多数出現しているが、テキスト型プログラミング教育においては、教師等の指導者とその学習環境を用意するために、環境を整備するうえでの知識が必要であり、困難であると考えられる。

本報告では、テキスト型プログラミング教育を行う指

導者が、容易かつ低廉に、多数の学習者に同時に共通のプログラミング学習環境を提供できるシステム(以下、本システムという)をシングルボードコンピュータRaspberry Pi上に開発したので、この詳細について述べる。また、学習者側においても、すでに所有している機器・端末等を持ち寄り、有効に生かしながら学習できるBYOD(Bring Your Own Device)によるシステムを目指し、その動作状況を確認した。

なお、本システムの学習対象者は、テキスト型プログラミング言語の使用ならびにOSおよびテキストエディタ操作の技能を要するため、中学生以上一般利用者としている。

2 システムに必要な機器・ソフトウェア

図1に、システム利用状況の全容を図式化したものを示す。図1の本システムは、シングルボードコンピュータRaspberry Pi Zero W⁴⁾にGPIOピンヘッダ(40ピン)を追加したRaspberry Pi Zero WH上に構成される。Raspberry Pi Zero WHを使用する理由を以下に示す。

(理由1)Linux系オペレーティングシステムRaspbianが使用可能であること

^{*1} 岐阜大学情報連携統括本部

Development of Computer Programming Learning Environment by Using Single-board Computer

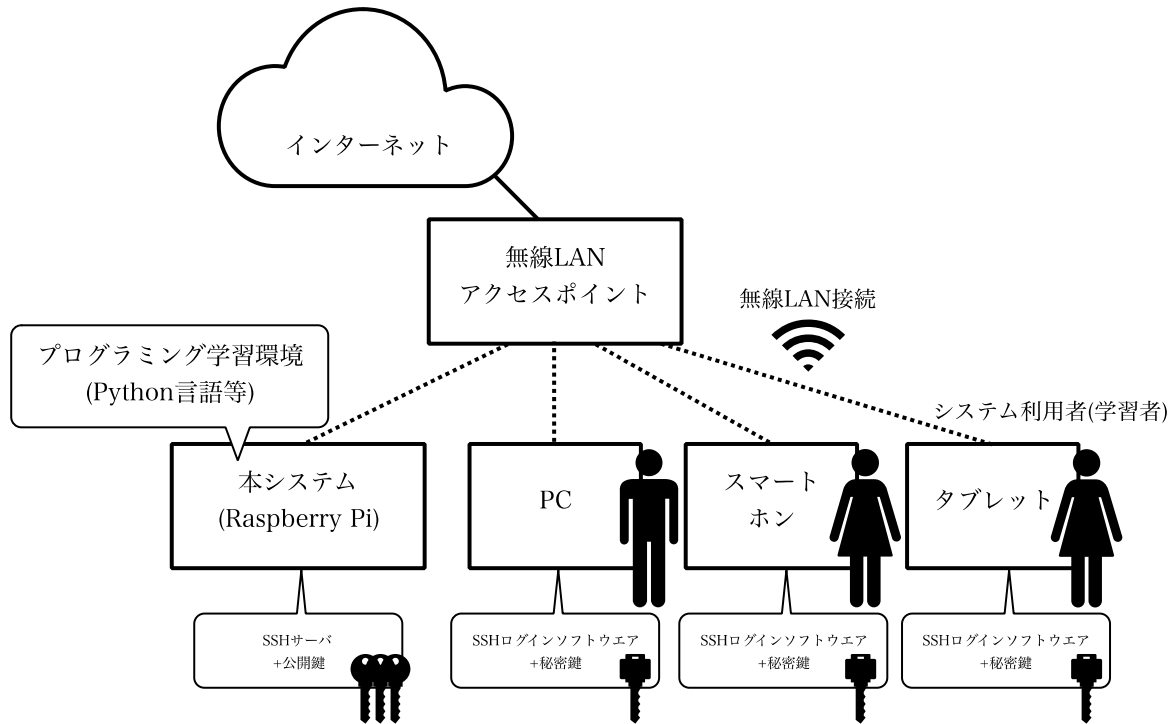


図1 本システムの概要

Raspbianは、Debian系ディストリビューションのLinuxをベースとしたRaspberry Piに正規に対応するオープンソースのオペレーティングシステム(OS)であり、多数のソフトウェア資産が無償で提供されている。またプログラミング教育を実施する上で不可欠なプログラミング言語も多数存在する。さらに、プログラム開発時に必要となるテキストエディタ等のツールも豊富に用意されており、関連情報もインターネット・書籍等で公開・共有されている。

(理由2)軽量・低廉・低消費電力であること

Raspberry pi Zero WHの大きさ・重量は65mm×30mm・10グラム程度と小型軽量であり、可搬性に優れる。ただし、シングルボードコンピュータの性質上基板・端子等がむき出しであるので、小型のプラスチックケースなどに入れることが望ましい。

Raspberry Pi Zero WHはRaspberry Pi Zero WにGPIOピンヘッダ(40ピン)が実装(はんだ付け)されたもので、国内で2,000円弱程度で販売されている。GPIOピンヘッダの付属しないRaspberry Pi Zero Wはこれよりやや安価であるが、販売店によっては本体単体での購入個数制限(1人あたり1台まで)が設けられていることがある。Raspberry

Pi Zero WHにおいてはそのような制限がないことが多い。

電源は5V DCをUSB Micro-BコネクタもしくはGPIO端子から給電する。消費電力は0.75Wとされており、市販の2.1A出力・電池容量3200mAh程度のモバイルバッテリーなどからUSB給電することで数時間連続稼働可能である。

(理由3)無線LAN機能を有すること

Raspberry Pi Zero WHは本体にIEEE 802.11 b/g/n の無線通信機能を有しているため、他の端末等から無線LANを用いたネットワーク接続によるリモートアクセスを行うことが可能である。また、リモート操作することにより、運用時にあたってディスプレイ、キーボードなどの入力機器を省くことができる。

なお、上記(理由1)、(理由3)を満たすRaspberry Piであれば、本システムはどの機種でも利用可能である。

次に、本システムにネットワーク接続してプログラミング学習を行う学習者側の端末等について述べる。端末については、無線LAN機能を有するPC(Windows, Mac)のほか、一般的に普及しているスマートホン、タブレットPC等を使用することが可能である。これにより、学習時に利用者が自身の所有する接続端末を持ち寄り(BYOD)、本シス

テムが提供するプログラミング学習に参加することが可能である。

本システムにアクセスするためには、端末等にSSHログイン接続をするソフトウェアの実装が必要である。代表的な端末等において、利用検証を行い、動作確認できたSSHログインソフトウェアの代表例を示す。

- a) Windows 10: TeraTerm(フリーソフト)⁵⁾
- b) macOS: ターミナル(OS 標準)
- c) iOS(iPhone, iPad 等), Android: Termius(無償アプリ, App Store や Google Play からインストール可能)⁶⁾

以上のほかでも、有償ソフトウェア・有償アプリ等でSSH接続が可能かつ公開鍵認証に対応しているものは本環境で利用できるものと思われる。なお、上記 c) のTermiusについて、画面タップによる仮想キーボードで一応操作可能であるが、Bluetooth等で接続できる物理キーボードを用意することが望ましい。

3 システムの開発・設定

本システムにおいて、運用上最低限必要とされる機器等は次の三点である。

- ・Raspberry Pi Zero WH
- ・microSD カード(8GB 以上)
- ・USB Micro-B コネクタによる電源入力(AC アダプタ, USB 電源アダプタ, モバイルバッテリー等)

このほか、無線 LAN 接続環境(無線 LAN ルータの SSID, パスフレーズ), 設定用 PC 等を用意する。

まず、microSD カードに、OS である Raspbian をインストールする。Raspberry Pi 財団の公式 Raspbian ダウンロード Web サイト⁶⁾においては、現在三種類のダウンロードイメージが提供されているが、今回のシステムにおいては、これらのうち最もファイルサイズの小さい Raspbian Buster Lite(September 2019/435MB)を用いる。続いて、ダウンロードした OS イメージを書き込んだ microSD カードを PC に接続し、microSD カード内のファイルシステムにおいて /boot/ssh という空ファイルを作成する。また、/boot/wpa_supplicant.conf というファイル名で下記のような内容(一例)のファイルを作成しておく。これは、本システムを既存の無線 LAN ルータ(親機)に接続するための子機としての設定であるため、現在

運用されている無線 LAN 環境において適宜変更する必要がある。

```
country=JP

ctrl_interface=DIR=/var/run/wpa_supplicant

GROUP=netdev

update_config=1

network={

    ssid="既存の無線 LAN ルータの SSID"

    psk="無線 LAN ルータのパスフレーズ"

}
```

これにより、Raspbian 起動時に既存の無線 LAN ルータに Raspberry Pi が接続され、SSH 接続により外部からアクセス可能になる。なお、Raspberry Pi Zero WH の無線 LAN は 2.4GHz 帯のみに対応しているため、無線 LAN ルータ側の運用周波数に注意する必要がある。

上の設定ののち、microSD カードを PC から取り外し、Raspberry Pi に取り付ける。続いて USB Micro-B コネクタに電源を接続し、起動する。本システムは Raspberry Pi をヘッドレス(操作のための入出力機器を接続せず、本体と電源のみで稼働すること)にて運用するので、ディスプレイやキーボードを接続する必要はないが、稼働状況をモニタしたり、ネットワークを介さず直接 Raspberry Pi を操作したい場合は、ディスプレイ等を接続しても良い。

Raspberry Pi を起動して 2~3 分後、Raspberry Pi が接続している無線 LAN ルータと同一のネットワークに接続された PC 等の端末の SSH ログインソフトウェアから、初期設定時のみ下記のコマンドを入力して Raspberry Pi にアクセスする。

```
ssh pi@raspberrypi.local
```

ログインプロンプトが表示されれば、接続は完了である。続いて、ネットワークの設定(スタティック IP アドレスの設定)、本体ソフトウェアのアップデート、およびプログラミング学習環境のためのプログラミング言語のインストールなどを行う。今回のシステムでは、プログラミング言語として Python 3.7⁸⁾を導入した。さらに、SSH サーバとなる Raspberry Pi にてユーザごとに SSH 鍵認証鍵ペアの生成を行い、公開鍵を Raspberri Pi に設置する。また、事前に秘密鍵を学習者の端末(PC 等)に事前に設置しておく。パスワード認証を用いずに公開鍵認証を用い

る理由は、通信の安全性を保つのみならず、ユーザ ID とパスワードの入力が不必要となるため、ログイン時の操作の簡略化となる。

4 システムの利用

本システムを利用している様子を図 2 に示す。図 2 は、

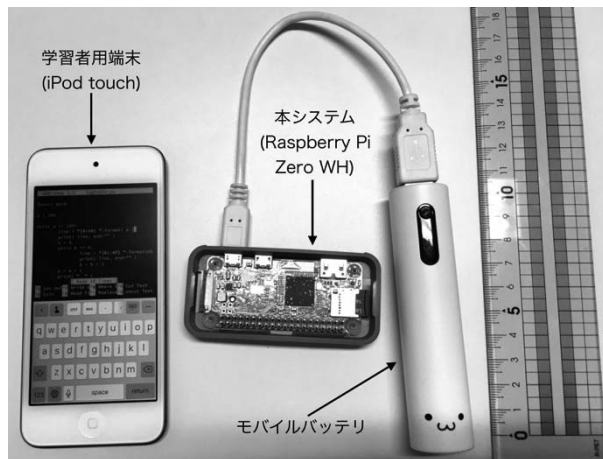


図 3 本システム運用中の様子



図 2 本システムを用いた Python プログラミング

プラスチックケースに収めた Raspberry Pi Zero WH をモバイルバッテリーで稼働し、学習用端末である Apple iPod touch から無線 LAN ルータを介して Termius でログインし、GNU nano エディタで Python プログラムを編集している状態である。なお、無線 LAN ルータはこの図には写っていない。図 3 は利用者のターミナルソフトウェア (Termius) における Python 言語のプログラム (常用対数表の作成) 編集中の iPod touch (iOS) のスクリーンショットである。学習者の端末として、このほか Windows の TeraTerm, macOS のターミナル, Android の Termius, iPad の Termius からのログインを確認した。いずれも、認証には端末に秘密鍵を設置した公開鍵認証を使用している。

プログラム入力・編集にはテキストエディタ GNU nano9) を使用している。Raspbian においては、テキストエディタとして vi 系エディタ (vim 等) またはその他のソフトウェア (Emacs 等) を使用できるが、ESC (エスケープ) キーを多用せずに操作でき、動作も軽量である GNU nano がふさわしいと思われる。スマートフォン・タブレット端末に接続するキーボードには ESC キーがないことが多く、画面上の仮想 ESC キーを表示させて押下することが多い。仮想 ESC キーは操作性が良いとはいえないため、ESC キーを多用する vi 系エディタは不便である。また、他のエディタはインストールは可能なものの、処理性能を抑えた Raspberry Pi Zero WH への負荷が高いものもあるため、軽量のテキストエディタである GNU nano が適する。

本システムにおいては、テキスト型プログラミング教育を可能とするものとするため、GUI 等の設定は不要である。

5 おわりに

本来 Raspberry Pi をはじめとする一部のシングルボードコンピュータは、プログラミング教育の導入のために開発されたという経緯がある。しかしながら、シングルボードコンピュータ自体の特性として、本体にディスプレイやキーボードなどの入出力装置を持たず、利用者がそれらを自身で用意する必要がある、それらを接続することで可搬性が損なわれるなどのデメリットも避けられな

かった。Raspberry Pi をノート PC のように運用できるサードパーティ機器も存在するが、比較的高価であり、キーボードの操作性も良好であるとは言えず、利用者が自身で組み立てる必要があるなど、利用の敷居は低いとはいえなかった。また、Raspberry Pi そのもののプログラミング環境設定も、Linux 等 UNIX 系システムが不慣れた教育者には、OS のインストールからその整備に至るまで相当な苦勞を伴うものと思われる。

本システムは、そういった入出力に関わる部分を利用者が BYOD として持ち寄らせることで、自分が使い慣れた端末でプログラミングできるものである。また、本システムはディスプレイやキーボードを接続しないヘッドレス運用であるので、可搬性に優れる。また、本システムの電源等の確保も AC 電源だけでなくモバイルバッテリーなども利用できるため、指導者は場所を問わずプログラミング学習環境を提供できる。さらに、グラフィックス等を用いないテキストベースの環境とあって、性能を抑えて安価に販売されている Raspberry Pi Zero WH でも問題なく動作させることができた。

バックアップやシステムの配布等についても、SD カードイメージをまるごとファイルとして保存すればよいので、Raspberry Pi もしくは SD カード破損時の回復も容易であり、他への配布も可能である。

本システムにおいて使用したオペレーティングシステムやソフトウェアは、極力無償で入手できるものを利用した。また、学習者側のソフトウェアについても、オープンソースもしくは無償で入手できるものである。したがって、プログラミング学習における導入コストを削減することができる。

現在の課題として、本システムでは、テキスト型プログラミング言語をテキストエディタで入力し、実行する上で、UNIX 系 OS の初歩的な操作とテキストエディタの利用をまず習得する必要がある。また、直接グラフィックス表示はできないので、学習するプログラミング内容に制限がある。こういった課題を解決する上で、Jupyter notebook 等の Web ベースのプログラミング学習環境を導入することを検討している。また、現時点で無線 LAN ル

ータを介して本システムに接続するが、本システムを無線 LAN ルータ化することで、別途無線 LAN ルータ等を必要としないシステムに拡張することも考えられる。また、学校等で本システムを運用する場合、既存の PC およびネットワーク設備との混在・整合性に配慮することが必要であろう。

引用・参考文献

- 1) 文部科学省(2018) 小学校プログラミングの手引(第二版)
https://www.mext.go.jp/component/a_menu/education/micro_detail/_icsFiles/afieldfile/2018/11/06/1403162_02_1.pdf (2019 年 12 月閲覧)
- 2) 松澤芳昭, 酒井三四郎(2013) ビジュアル型言語とテキスト記述型言語の併用によるプログラミング入門教育の試みと成果, 情報処理学会研究報告コンピュータと教育(CE), Vol.2013-CE-119 No.2.
- 3) Scratch
<https://scratch.mit.edu> (参照日 2019 年 12 月 24 日)
- 4) Raspberry Pi Zero W
<https://www.raspberrypi.org/products/raspberry-pi-zero-w/> (参照日 2019 年 12 月 24 日)
- 5) Tera Term Open Source Project
<https://ttssh2.osdn.jp/index.html.ja> (参照日 2019 年 12 月 24 日)
- 6) Termius
<https://termius.com> (参照日 2019 年 12 月 24 日)
- 7) Download Raspbian for Raspberry Pi
<https://www.raspberrypi.org/downloads/raspbian/> (参照日 2019 年 12 月 24 日)
- 8) Welcome to Python.org
<https://www.python.org> (参照日 2019 年 12 月 24 日)
- 9) GNU nano
<https://www.nano-editor.org> (参照日 2019 年 12 月 24 日)