

中学校技術・家庭科(技術分野)における Python を用いた計測・制御システムのプログラミング環境の構築と利用

安田慎吾^{*1}, 山田哲也^{*2}, 伊藤陽介^{*3}, 阪東哲也^{*4}, 曾根直人^{*3}, 藤原伸彦^{*5}, 長野仁志^{*6}

平成 29・30 年に告示された新学習指導要領ではプログラミング教育に関する内容が充実した。本論文では中学校から高等学校へのプログラミング教育の継続性に着目し、計測・制御システムに用いるプログラミング言語として Python の採用を提案する。中学校技術・家庭科(技術分野)の学習内容である「生活や社会における問題を計測・制御システムのプログラミングによって解決する活動」において Python を利用できる教材を選定するとともに、高機能テキストエディタによるプログラミング環境の利用方法とプログラミングを段階的に学習できるように配慮したテキスト構成とプログラム例を示した。

[キーワード: 学習指導要領, 計測・制御システム, プログラミング, Python]

1. はじめに

近年、移動通信システムの高速大容量化や携帯端末の小型化・高性能化によりインターネットを介した様々なサービスが普及するとともに、社会のあらゆる分野において情報システムの利用が加速度的に拡大しつつある。このような社会情勢を鑑み、未来を担う子供たちを育成する視点で従来の学校教育で取り扱っていた情報に関する学習内容のうち「情報の科学的な理解」をより充実させることが求められ、平成 29・30 年に告示された小学校、中学校及び高等学校学習指導要領(以下、新学習指導要領)ではプログラミング教育に関する内容を充実させている[1~3]。

とくに、中学校学習指導要領技術・家庭科(技術分野)(以下、技術科と略記)の内容「D 情報の技術」では、双方向性のあるコンテンツのプログラミング、及び計測・制御システムのプログラミングが規定され、令和 3 年度から全面実施される。中学校におけるプログラミング教育は、小学校との繋がりに加えて高等学校への繋がりも考慮しなければならない。

令和 4 年度から新学習指導要領に移行する高等学校情報科の科目「情報 I」に対応した教員研修用教材が平成 31 年 3 月に文部科学省より公表された[4]。この教材で採用されているプログラミング言語は、Python, JavaScript, VBA, ドリトル, swift であり、すべてが文字を組み合わせてプログラムを構成するテキスト型プログラミング言語である。これらのうち汎用プログラミング言語の一種である Python は 2021 年 2 月時点で ISO や JIS で規格化されていないものの、単純な文法や予約語の少なさ、豊富なライブラリの提供に加えて多機能な統合型開発環境が利用できることから、産業用のみならず教育用を含めた幅広い業界で国際的に普及している。

一方、技術科における計測・制御システムのプログラミングの実施学年について既習事項を考慮すると中学校 2 年後半または 3 年に実施されることが想定される。そのため、本論文では中学校から高等学校へのプログラミング教育の継続性に配慮し、技術科の学習内容である「生活や社会における問題を計測・制御システムのプログラミングによって解決する活動」において Python の利用を提案するとともに、利用できる教材を選定し、プログラミング環境及び新たに作成したプログラミング用テキストについて述べる。

2. 計測・制御システムを構成する教材

生活や社会における問題を発見し、計測・制御システムを構成して課題解決を図る教材は、当該システムの物理的な構成を容易に変更できるとともに、プログラムによってシステムを稼働させることがで

^{*1} 鳴門教育大学大学院 高度学校教育実践専攻 自然・生活系教科実践高度化コース(技術・工業・情報科教育実践分野)大学院生

^{*2} 鳴門教育大学 附属中学校

^{*3} 鳴門教育大学大学院 高度学校教育実践専攻 自然・生活系教科実践高度化コース(技術・工業・情報科教育実践分野)

^{*4} 鳴門教育大学 情報基盤センター

^{*5} 鳴門教育大学大学院 高度学校教育実践専攻 教員養成特別コース

^{*6} 鳴門教育大学 附属小学校

きななければならない。さらに新学習指導要領では持続可能な社会を目指すことも規定されており、当該システムの構成物をすべて再利用できることが望ましい。

以上述べた点を考慮した結果、コンピュータを内蔵した部品やギアなどの機構部品、一般部品を規格化された格子状の筐合部を用いて容易に脱着して計測・制御システムを構成できるブロック教材である LEGO Mindstorms EV3（以下、EV3 と略記）を採用する。部品同士の筐合は摩擦力のみで維持されるため、学習を終えた後作品を分解することで全部品を再利用できる。表 1 に EV3 の部品構成を、表 2 にコンピュータを内蔵している EV3 ブロックの主な仕様をそれぞれ示す。

3. 計測・制御システムのプログラミング

環境

EV3 ブロックの標準ファームウェアで起動する OS は Python によるプログラミングに対応していない。そのため、Debian 系 Linux ディストリビューションをカスタマイズした ev3dev と呼ばれる OS を用いる [5]。ev3dev のイメージファイルをパソコンなどで書き込んでおいたメモリカード (microSDHC, 4~32G バイト) を EV3 ブロックに装着した状態で電源を投入すると標準ファームウェアの代わりに ev3dev が起動される。図 1 に EV3 ブロックの LCD に表示される ev3dev のメニュー例を示す。メニューは上下左右などのボタンで操作する。

ev3dev には、python-ev3dev と呼ばれる Python 用ライブラリがインストールされており、EV3 が備える各種センサによる計測及びモータや LED などの制御を可能にしている [6]。python-ev3dev に含まれているモジュールの一部を表 3 に示す。

EV3 ブロックとパソコンの接続は、USB ケーブルによる有線や Bluetooth による無線で行う。EV3 ブロックに無線 LAN 用 USB ドングルを装着することで無線 LAN 接続も可能である。各接続方法の特徴や注意点を表 4 に示す。

ev3dev のデフォルトのホスト名は「ev3dev」であり、同時に複数の EV3 ブロックを使って無線接続する場合、識別できなくなる。そのため、パソコン側から ssh 対応のターミナルソフトウェアを使って EV3 ブロックにログインし、管理者権限で「ev3dev-config」コマンドを実行する。その後「4 Advanced Options」→「A1 Hostname」を選択し、個別のホスト名を設定しておく。

また、パソコン側 OS で Windows を利用し、USB

表 1 EV3 の部品構成

種別	個数
EV3 ブロック	1
サーボモータ (ポテンショメータ内蔵)	3
接触センサ	2
光センサ	1
超音波距離センサ	1
ジャイロセンサ	1
機構部品・一般部品など	532

表 2 EV3 ブロックの主な仕様

	規格など
OS	Linux
CPU	ARM9 300MHz (32bit)
メモリ	RAM 64M バイト フラッシュメモリ 16M バイト
表示	LCD 横 178 ピクセル, 縦 128 ライン LED 2 個
ボタン	中央, 上, 下, 左, 右, 戻る
インターフェース	出力 4 ポート, 入力 4 ポート USB コネクタ (Mini-B, Type A)
発音	スピーカー
通信	USB2.0, Bluetooth, Wi-Fi (USB ドングル)
カードスロット	microSD カード (microSDHC, 最大 32G バイト)
ソフトウェア	EV3 ソフトウェア, C, Java, Python, C#, mruby, LabVIEW, Scratch

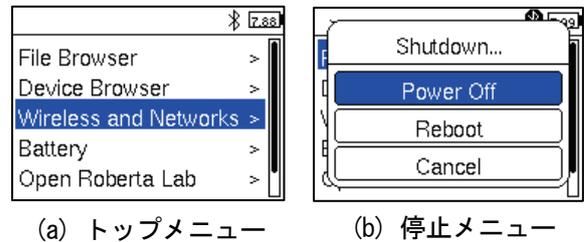


図 1 ev3dev のメニュー例

表 3 python-ev3dev のモジュール (抜粋)

モジュール名	主な機能
ev3dev2.button	ボタン状態の読み込み
ev3dev2.console	LCD にコンソール表示
ev3dev2.display	LCD に図形描画
ev3dev2.fonts	フォントの読み込み
ev3dev2.led	LED 制御
ev3dev2.motor	モータ制御
ev3dev2.port	入出力ポートの再設定
ev3dev2.power	電源状態の読み込み
ev3dev2.sensor.*	各種センサによる計測
ev3dev2.sound	発話や音の発生
ev3dev2.wheel	車輪やタイヤの直径

表 4 EV3 ブロックとパソコンの接続方法

接続方法	長所	短所	注意点
USB ケーブル	<ul style="list-style-type: none"> 接続中の EV3 ブロックを USB ケーブルで確認できる。 パソコン側がインターネット接続されていると「ブリッジ接続」設定することで、EV3 側からインターネット接続できる。 	<ul style="list-style-type: none"> USB ケーブルを装着したままロボットを動作させにくい。 USB ケーブルの脱着に時間がかかる。 EV3 ブロックの USB コネクタを劣化させる。 	<ul style="list-style-type: none"> EV3 ブロックの USB コネクタを劣化させないように延長用コネクタを装着しておくことが望ましい。
Bluetooth	<ul style="list-style-type: none"> 約 10m の見通し距離までであれば、パソコンから離れていてもプログラムを更新したり実行したりできる。 パソコンと一度ペアリングすることで、EV3 ブロックを識別できる。 	<ul style="list-style-type: none"> 約 10m の見通し距離までしか通信できないため、広い教室では対応できない場合がある。 通信が不安定になり切れてしまうことがある。 パソコンとのペアリング設定が解除されると、再度ペアリングする手間がかかる。 	<ul style="list-style-type: none"> パソコン側に Bluetooth 機能を備えている必要がある。 EV3 ブロックを識別するための名称を ev3dev に設定する必要がある。
無線 LAN	<ul style="list-style-type: none"> 約 50m の見通し距離までであれば、パソコンから離れていてもプログラムを更新したり実行したりできる。一般的な教室であれば、無線 LAN でほぼ対応できる。 通信状態が保持されやすい。 インターネットに接続できる。 	<ul style="list-style-type: none"> EV3 ブロックに無線 LAN 用 USB ドングルを追加で装着しなければならない。 約 50m 程度の見通し距離までしか通信できない。 SSID による認証であれば問題ないが、他の認証方式では対応できない場合がある。 アクセスポイントを設置しなければならない。 	<ul style="list-style-type: none"> パソコン側に無線 LAN 機能を備えている必要がある。 EV3 ブロックで利用できる無線 LAN 用 USB ドングルは ev3dev で認識可能なものに限られる。 EV3 ブロックを識別するため、固定 IP アドレスを ev3dev に設定する必要がある。

ケーブルや Bluetooth によって接続している場合、パソコンと EV3 ブロックにそれぞれ固定 IP アドレスを設定することで接続時間を短縮できる。

パソコン側で作成したり修正したりした Python のプログラムは EV3 ブロックに送信され実行される。その際、プログラミング及び送信、実行操作などを容易にするため多機能テキストエディタの一種である Microsoft 社が開発・提供している Visual Studio Code (VS Code) を用いる。

VS Code をパソコンにインストールした後、メニューを日本語化する。Python で記述されたプログラムの文法チェックやコード補完機能を行うため「Python extension for Visual Studio Code」という拡張機能を VS Code にインストールする。さらに、EV3 ブロックとの通信も行うため「MicroPython for LEGO MINDSTORMS EV3」もインストールする。その際、EV3 ブロック内のファイルやディレクトリの一覧を表示する拡張機能「ev3dev-browser」もインストールされる。

VS Code を起動し、EV3 ブロックとパソコンを接続後、作成したプログラムの実行状態を示す画面例を図 2 に示す。パソコン側で作成したファイルやディレクトリ構成は「EV3DEV DEVICE BROWSER」メニューを操作することで、容易に EV3 ブロックに転送できるとともに、特定のファイルを指定してプログラムを実行できる。実行開始後、VS Code のウィンドウ内に操作アイコンがポップアップされ、プログラムの停止や再開などができる(図 3)。

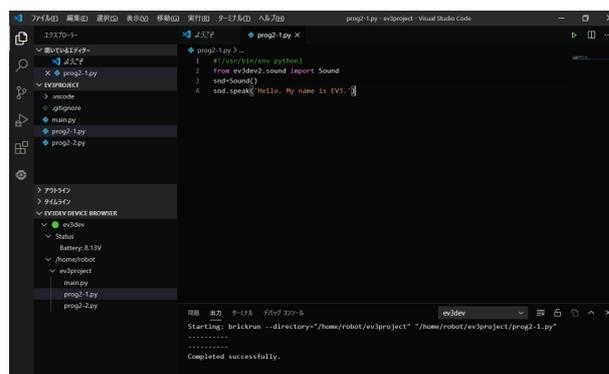


図 2 VS Code の画面例

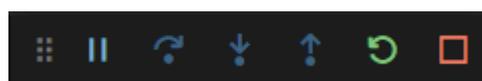


図 3 操作アイコン

4. プログラミング用テキストの構成

Python による計測・制御システムのプログラミングを学習するためのテキストは、中学校段階の学習者が興味・関心をもって習得できるように配慮し構成されている[7]。EV3 を用いることで自由度の高い計測・制御システムを試作したり、改良したりできるが、本テキストではプログラミングの際に必要な基本的な事項を段階的に学習できるように、対象とする EV3 の形状を限定して説明している。具体的な形状は、EV3 ブロック単体、EV3 ブロックと接触センサ、車輪移動型ロボット(図 4)の 3 種類とし、それぞれに対応するプログラム例を掲載している。

本テキストでは、サーボモータ、LCD、LED の制御



図4 車輪移動型ロボットの外観

処理、及び接触センサ、光センサ、超音波距離センサ、ポテンショメータによる計測処理を取り扱う。なお、中学校教育の内容を考慮し、本テキストではジャイロセンサは取り扱わず、ポテンショメータによる計測値は回転数に置き換えて説明している。

また、「チャレンジ」と名付けた問題を解決するプログラムを自分で作成することで、今まで学んだことがわかっているかどうか確認できるようにしている。

まず、EV3ブロックとVS Codeの操作方法を学習する。その際、英文を発話させるプログラム例(図5)を入力し実行することで興味・関心を高める。Pythonではインデント(段付け)が表記上重要であるため、空白は「□」と記載し明確に判読できるように配慮している。LCDに線や点、円で簡単な絵を描画させたり、画像を表示させたりするプログラムを示し、Pythonの基本的な記述方法を説明している。図6(a)にLCDへの図形描画プログラム例を示し、その実行結果を図6(b)に示す。

つぎに、EV3ブロックに接触センサを取り付けたものを使って計測した結果を処理するプログラムについて学習する。スピーカーを制御対象として音階を使って音楽を演奏させたり、音声ファイルを使って再生したりするプログラムを実行しつつ、音階を例として変数の使い方も学習する。

車輪移動型ロボットを用いて逐次処理(図7)から始めて条件判断処理や反復処理(図8)を行うプログラムを順に説明している。ロボットの動きとプログラムの内容との関係性を認識させることで処理方法をより深く理解できる。計測・制御システムの例示としてラインをたどる「ラインレース・ロボット」を取り上げている。ここでは、光センサに付随したLEDを光らせ、その反射光の強弱によってラインの有無を判別する方法やラインから外れた場合の探索方法などのアルゴリズムを説明している。

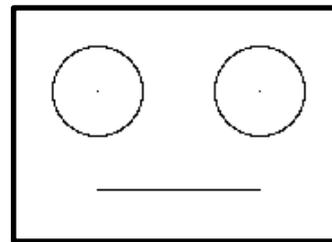
本テキストを用いてPythonによるプログラミング

```
from ev3dev2.sound import Sound
spk=Sound()
spk.speak('Hello._Nice_to_meet_you.')
```

図5 発話するプログラム例

```
from time import sleep
from ev3dev2.display import Display
lcd=Display()
lcd.clear()
lcd.point(False,45,45)
lcd.point(False,135,45)
lcd.circle(False,45,45,25,fill_color=None)
lcd.circle(False,135,45,25,fill_color=None)
lcd.line(False,45,100,135,100,width=1)
lcd.update()
sleep(10)
```

(a) プログラム



(b) 実行結果

図6 LCDへの図形描画プログラム例と実行結果

```
from time import sleep
from ev3dev2.motor import MoveTank,OUTPUT_B,OUTPUT_C
rbt=MoveTank(OUTPUT_B,OUTPUT_C)
rbt.on(30,30)
sleep(4)
rbt.on(-30,-30)
sleep(4)
rbt.off()
```

図7 ロボットを移動させるプログラム例

```
from ev3dev2.motor import MoveTank,OUTPUT_B,OUTPUT_C
rbt=MoveTank(OUTPUT_B,OUTPUT_C)
sp=20 # スピード(%)
rt=0.54 # 90°ターンに必要な回転数
len=300 # 前進する距離(mm)
d=56 # タイヤの直径(mm)
rs=len/(3.14*d) # 前進に必要な回転数
for i in range(4):
    rbt.on_for_rotations(sp,sp,rs) # 前進
    rbt.on_for_rotations(sp,-sp,rt) # 右ターン
rbt.off()
```

図8 ロボットを移動させるプログラム例

について習得した後、EV3 を用いて課題解決を図るための計測・制御システムを試作したり、改良したりする。技術科におけるプログラミング教育は、プログラマ養成ではないため、学習者がPython で記述されたプログラムのすべてを入力するのではなく、あらかじめ指導者側で一部を空白としたプログラムのファイルを準備しておき利用する授業方法が有効と考えられる。

5. まとめ

生活や社会における問題を計測・制御システムのプログラミングによって解決するための技術科の学習活動で活用できる教材に関して、試作や改良の容易性ならびに持続可能な社会の構築に繋がることなどを考慮し教材を選定した。この教材に対応するPython のプログラミング環境の一種であるVS Code の操作方法と複数台数の教材を無線接続して利用する場合の設定などについて述べた。Python による計測・制御システムのプログラミングについて段階的に学習できるように配慮したテキストの構成とプログラム例を示した。

令和 2 年度に徳島県内 F 中学校において 16 台のEV3 とパソコンをBluetooth でそれぞれ無線接続し、本テキストの一部を用いた研究授業を実施した。その結果、EV3 を用いた計測・制御システムのプログラミングにPython を利用できる可能性が示唆された。

今後、様々な課題解決に対応するための計測・制御システムを構築できるようにプログラミング用テ

キストにPython の関数や並列処理などについて追記していく予定である。

参考文献

- [1] 文部科学省(2017) 小学校学習指導要領(平成 29 年告示)解説総則編, 東洋館出版社.
- [2] 文部科学省(2017) 中学校学習指導要領(平成 29 年告示)解説技術・家庭編, 開隆堂出版.
- [3] 文部科学省(2018) 高等学校学習指導要領(平成 30 年告示)解説情報編, 開隆堂出版.
- [4] 文部科学省(2019) 高等学校情報科「情報 I」教員研修用教材, https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1416756.htm (最終アクセス日: 2021 年 2 月 27 日).
- [5] ev3dev, <https://www.ev3dev.org/> (最終アクセス日: 2021 年 3 月 1 日).
- [6] R. Hempel, et al. (2020) python-ev3dev Documentation, Release 2.1.0.post6, <https://media.readthedocs.org/pdf/python-ev3dev/latest/python-ev3dev.pdf> (最終アクセス日: 2021 年 2 月 27 日).
- [7] 伊藤陽介(2020) プログラミング言語 Python による計測・制御入門, <https://www.naruto-u.ac.jp/facultystaff/ito/research-paper-public/2020/python-ev3-textbook.pdf> (最終アクセス日: 2021 年 3 月 1 日).