

# Raspberry Piを用いた中学校教員養成における計測・制御プログラミング環境の構築と実践

鎌田 敏之<sup>1</sup>

**概要:** Raspberry Pi と教育用プログラミング言語「ドリトル」による計測・制御プログラミングの学習環境を構築し、中学校技術・家庭科教員養成過程の学生実習に活用した。計測・制御の単元を学習するために、従来は PC とセンサ及びアクチュエータのインタフェースを USB 接続することが一般的であることに對し、本研究では将来の自律ロボット構築を見据え、無線通信を積極的に活用するとともに、Raspberry Pi の計算能力を活かし、「ドリトル」を同基板の上で実行し、PC からリモートデスクトップ接続することで、Raspberry Pi 内でデータの入出力を完結させることができるようにした。実践授業において、受講生は PC での「ドリトル」プログラム作成経験をそのまま活かすことができ、意欲的にセンサ及びアクチュエータを利用した作品制作に取り組む姿を観察することができた。

## Development and Practice of learning environment of "Measurement and Control" unit for teacher training with Raspberry Pi

TOSHIYUKI KAMADA<sup>1</sup>

**Abstract:** In this report, the development of specially customized Linux environment and interface PCB to communicate with sensors and actuators are described. The difference conventional learning environment of the "Measurement and Control" for teacher training course is the embracement of independent Linux board "Raspberry Pi" and specially made Arduino add-on board. By using this "wireless and headless" learning material with the student's own PC, they could continue to operate programming language "Dolittle" from on their PC with its screen to the display of remote desktop of Raspberry Pi running "Dolittle" on it smoothly, and they tried to finish their project with keeping engagement.

### 1. はじめに

中学校技術・家庭科の免許科目として学習指導要領に設定された「情報に関する技術」における、「プログラムによる計測と制御」に對する実習授業のために構築した学習環境と、その結果について報告する。

平成 20 年度に告示され、平成 24 年度より前面実施されている学習指導要領に對し、教員養成の場面で「計測と制御」を含むプログラム作成を行う実習授業に関する教材研究及び実践報告が多数なされている。（参考文献）このとき、教員養成を主な目的とする教育学部においては、教育現場である中学校での授業実施状況を想定し、PC 教室

に設置された PC を主として、それに USB 等で接続した入出力装置との通信、もしくは自律動作する MCU 基板へのプログラム転送によってセンサからデータを取得し、プログラムの処理にしたがって画面表示を操作したり、音やモータなどアクチュエータの操作を行う形式が多い。

著者は CE-99[1] で報告した通り、1998 年より MCU 基板を搭載した自律型ロボットカーを題材とし、課題達成のためゴールとしたロボットコンテストを通して計測・制御を伴うプログラムの作成を行わせる授業に取り組んできた。しかし、2004 年の教員免許法改正による教科に関する科目の単位数半減と卒業要件となる総単位数の削減により、PC 画面上で完結するプログラム作成の学習を半期で行い、計測・制御を伴うプログラムをそれに続く半期で実施する形態でプログラムを学ぶカリキュラム構成が徐々に

<sup>1</sup> 愛知教育大学  
Aichi University of Education

機能不全となった。具体的には、免許法上実習を含む単位履修が必須である点を利用し、前半のプログラム実習授業は必修を維持することができたが、それに続く授業は選択となったことで、セメスターあたりの取得単位に上限を設ける施策（いわゆる『キャップ制』）との関係から卒業単位数ギリギリの取得により卒業する学生が大半となり、年々受講生が減少していった。2010年度以後の受講生は、実質ゼロである。

一方で学習指導要領改訂により「プログラムによる計測と制御」は必修項目となっていることから、必修であるひとつの授業のなかに、プログラムそのものの学習と計測・制御を含む課題解決の両者を実施することが必要となった。これを実現するため、ロボットカーは基本的な組み立てを授業時間外もしくは教員が半完成品として短時間で完了できるよう用意し、受講生自身が取り組むプロジェクトは最終課題のみとすることとなった。

このように制約の多いなかで受講生の学習意欲を喚起し、目的とする「プログラムによる計測と制御」の基礎的かつ本質的内容の習得を達成するためには、学習における認知負担が低く、一方で原理から応用に至るまで、学習者が深い関心を持つ場合にも対応できる奥行きを提供する学習環境が必要となる。また、教員養成における必修授業であることから、免許科目である中学校技術・家庭科において、学校現場での実践場面での状況との乖離が少ないことが望ましい。

そこで、本報告ではPC上でのプログラム体験がそのまま自律型ロボットカーのプログラムまで、授業目的に沿った一貫した概念で実習を段階的に進められる学習環境の構築を行い、教職課程の大学生に対して試行を行った結果について述べる。また、試行的授業実践によって明らかになった課題とその対処方法についての考察も述べる。

## 2. 計測と制御を行うプログラムの学習状況

教員養成課程に入学する大学生の多くは、プログラム作成の経験をほとんど持たない。あったとしても、中学校または高等学校の授業のなかで、数時間の体験を行うに過ぎず、課題解決のためにコンピュータプログラムを活用することの具体的なイメージのない状態であるといえる。

日本における初等中等教育全体の教育課程のなかで、現行の学習指導要領においてプログラム作成について明示があるのは中学校技術・家庭科 [3] のみであるが、「情報とコンピュータ」は4分野のひとつでしかなく、また「計測と制御を行うプログラム」をなす6項目のうちのひとつであり、「コンピュータを利用したマルチメディアの活用」と並んで選択の扱いである。技術領域全体で実施可能な中学校3年間のうち85.25時間を単純に分割すると、「コンピュータを利用したマルチメディアの活用」を行わずに「計測と制御を行うプログラム」を選択したとしても、均等な時間

数をとるとして、計算上は4時間、2時間連続で時間割を組んだ場合は2回となる。

実際には必修項目は座学の要素が多いことから時間数を圧縮して「計測と制御を行うプログラム」の実習を行う時間は確保されるが、実際にプログラムにより計測と制御を行う学習には、制御対象に市販のキットを購入させることで製作を最小限にし、また「情報処理の手順を考える際に（中略）フローチャートなどを適切に用いる」とあることから、直接フローチャート部品をマウス操作で画面上に配置し、線で結ぶ操作により描いた図がそのままプログラムとしてUSB接続した基板上のMCUに書き込まれ、動作させる授業構成が、筆者が中学校における研究授業及び授業後の協議会で聴取した範囲での主流である。具体的には特定メーカーの特定教材がほぼ市場を独占しており、フォトリフレクタを2個用いたライントレース、前方バンパーの左右にある接触スイッチによる障害回避、赤外線LEDとフォトダイオード2組による非接触障害回避、もしくは環境光測定を用いた光の強さと移動を組み合わせた何らかの課題解決のいずれかを選択する。前者3つはデジタル値による分岐処理のみであり、最後の1つのみがアナログ値を用いた不等式による条件と対応した動作の組み合わせを実現する。

## 3. 教員養成における計測と制御を行うプログラム

公立中学校で本項目で授業研究を行う現職教員の方々は、こうした制約のなかで授業の目標設定を工夫することにより、生徒の意欲・関心を高めるべく努力されているが、それを支えるのは、生徒からの質問への対応や戸惑う生徒を支援する、教師自身の知見であり、自ら手を動かしてプログラムを作成できることができる自信と、センサーとアクチュエータがどのようにしてMCU回路とインタフェースをとり、プログラムのなかで表現されているデータが持つ意味はなにかという、本質的な理解であるべきと考える。教材のなかでブラックボックス化された回路とフローチャート描画による「手順を意図する動作に合致するよう配置する」ことのみを前提に、商品とともに用意されたワークシートを基準として何を加えるか、という工夫にとどまる限り、情報科学に向き合う態度はもとより、電気回路の理解も、なぜロボットカーが直進できるのかという疑問さえ問わない、あくまで「利用者」として情報機器に接する態度を超えることはなく、生徒がこの経験をもとに、コンピュータを利用した何かを生み出そうとする関心を深め、持ち帰った教材を発展的に利用しようと考えさせることは難しいと考えられる。

そこで、情報科学を体系的に学ぶ機会を持たない教員養成学部所属の学生が1つのセメスターで完結する実習授業のみで、上記の本質的理解に可能な限り近づけるため、

以下のようなシラバスを構成した。

### 3.1 シラバス

表 3.1 に示すのは、2014 年度前期愛知教育大学技術専攻 3 年生の必修授業として実施した、「プログラミング実習 II」のシラバスの要約である。受講者数は 15 名であり、そのうち男子学生 13 名、女子学生 2 名であった。

| 表 1 シラバスの要約 |                  |
|-------------|------------------|
| 時限          | 内容               |
| 1           | ドリトルの紹介、インストール   |
| 2           | タートルによる描画        |
| 3           | ボタンの利用           |
| 4           | アニメーション          |
| 5           | 簡単なゲーム           |
| 6           | 壁反射（ピンポンゲーム）     |
| 7           | シューティングゲーム       |
| 8           | 計測と制御のための電気回路    |
| 9           | マイコンボードの使い方      |
| 10          | 入力と出力            |
| 11          | ゲームコントローラーの作成と利用 |
| 12          | 自由課題の構想          |
| 13-15       | 自由課題の作成          |

前半の 7 回で、学生が入学時から持参している個人用ノート PC に、プログラミング言語「ドリトル」をインストールさせ、PC 画面を操作して、ドリトルの基本命令を順に覚えながら、意図したプログラムにどのように仕上げるかをテキスト [2] に沿って、またプログラムの設計・実装・テストとデバッグの手順について解説を加えながら理解させる内容を実施し、その後、センサーとアクチュエータの利用に関わる電気回路の講義を行った。同じ受講生は別の時限に「電気実習」を受講しており、トランジスタをドライバとして CdS 素子を入力として LED を点滅させる体験をしているため、オームの法則から始まる基本的な電気回路の知識と、MCU の入力端子が電圧を基準として状態もしくはアナログ値を得ることの説明は、比較的スムーズに行われた。

8 回目以後が今回報告する新たな試みとなるが、学習内容は、同テキストに掲載された「Arduino と通信しよう」の記述に沿ったものである。すなわち、「ドリトル」において Arduino を入出力装置として利用する際の「デジタル出力」、「アナログ出力 (PWM)」、「デジタル入力」、「アナログ入力」命令と、これらによって指定される I/O ピンに対する値の読み書きを「ドリトル」の画面によって確認することである。また、これによって、「計測」とは物理量を対象の単位で表される値としてコンピュータに与えられるのではなく、物理量を電気回路によって電気に関わる量（電流、抵抗、電圧）として観測し、コンピュータへは電圧に変換した上で入力端子に与えられること、また計測した

「データ」はあくまで相対的な「値」であり、デジタル入力であれば一定のしきい値電圧で 0 と 1 に切り分けられ、アナログ入力であれば、A/D コンバータの分解能で量子化された、電源電圧を基準とする入力電圧の相対値でしかないこと、また出力も同様に、コンピュータ内で 2 進法により表現された抽象的な「値」を、電圧として出力するにすぎないことを確認した。特に、いったんコンピュータ内の「数値」に抽象化されることで、他のデジタル表現と同じように、変換や通信、記録などが可能となることを強調した。技術を理解するための本質的な内容であるものの、実験で扱う事例を学習者が一般化して理解するには時間が足りないとの認識からである。

また、本実習授業では、2012 及び 2013 年度に、Arduino を用いてタクトスイッチ、アナログジョイスティック、CdS、LED を用いた計測と制御を実施したことを踏まえ、第 4 回から第 7 回までで取り扱ったゲームに、タクトスイッチとアナログジョイスティックからなるゲームコントローラ回路を接続するようにプログラムを書き換えることを行っている。また、2012 及び 2013 年度には、教員側が用意した単純なロボットカーのモータを H ブリッジによるモータドライバを用いて正逆転制御し、走行させる内容を 12 回目以後実施していたが、今回はモータ制御に限らない、自由な表現を行うプログラム作成のプロジェクトを与え、それにより最終試験とすることとした。

### 3.2 学習環境の設計

シラバスの範囲では、前年度までと同様、Arduino をノート PC に USB 接続してシリアル通信を行うことであっても、体験としては十分である。しかし、ゲームコントローラを自作した、という喜びは残るものの、「計測したデータを加工して制御する」ことの理解に至っていない様子がみられた。そこで、あえて PC 上のプログラムを離れ、新たに別のコンピュータを用意し、それが自作のプログラムによって計測と制御を自動実行していることを、その内部がわかる形で示すことで、より学習内容の意図を明確化することを考えた。

「Raspberry Pi」は Eben Upton 氏を代表とする「Raspberry Pi Foundation」によって設計と開発が行われ、大量生産によって 22 ポンドという価格ながら ARM アーキテクチャを持つ GPU 内蔵アプリケーションプロセッサ (SoC) を搭載した、Linux と X Window が稼働するカードサイズのワンボードコンピュータである。2014 年前半の時点では、「Raspberry Pi Type B」が 512MB RAM、USB 端子を 2 つ、100Base-T Ethernet 端子、HDMI 出力、ステレオオーディオ出力端子、コンポジットビデオ出力端子を備え、標準サイズの SD カードによって起動する、本実習に適切と考えられたモデルであった。現在ではコンポジットビデオ端子を省略し、USB を 4 つに仕様変更した「Type B+」

に置き換えられているところである。「Raspberry Pi」はまた、ピンヘッダにより 26 ピンの GPIO 端子を基板上に実装しており、これを通してデジタル入出力を行うことができるため、Linux 上のアプリケーションから多様なデジタルデバイスを接続したプロトタイピングが容易である。

しかし、Raspberry Pi の GPIO は搭載されている ARM SoC の I/O 端子そのものであるため、不注意により CPU の破壊を招きかねず、ここに LED やスイッチ等を直結するのは安全上望ましいとはいえない。また、A/D コンバータは SoC にもボード上にも含まれていないため、外付けが必要である。そこで、ラッチ IC やトランジスタアレイによるドライバ等を搭載し、GPIO を保護する拡張基板を試作したが、部品点数の多さから、価格の問題と製作の困難さが予想されたため、この方針はとらないこととした。また、「ドリトル」は Java で実装されているものの、Raspberry Pi の GPIO と直接通信するための Java ラッパーを言語から呼び出すには言語処理系の拡張を行う、もしくは既存の TCP/IP 通信のサーバを経由するなど変更の工数がかかることから、GPIO 上に出ている UART ポートを利用して、シリアル通信を行う MCU 基板の搭載がコスト及び工数の点で有利と考えた。

ここで利用する MCU は安価な 8bit マイコンでよい。であるなら、200 円程度で入手可能な Atmel 社の ATmega328P を利用し、Arduino 互換とすることで、ドリトルが用意する Arduino 用スケッチと Arduino 用命令定義がそのまま利用でき、テキスト通りの授業を行うことができる。ただし、Arduino の仕様のうち、実習に必要な最小限のところまで回路を省略することで、製作工数の削減も図ることとした。

### 3.3 製作した拡張基板

図 1 が、製作した拡張基板を Raspberry Pi に実装した状態を示した写真である。向かって左上に Raspberry Pi の GPIO 端子があり、拡張基板の裏側にあるソケットによって結合している。Arduino 互換ピン位置を持つユニバーサル基板を用いており、Arduino と同じ位置にソケットが配列されている。ただし、ISP 用の 6 ピンヘッダは省略したためファームウェアの書き換えはできない。逆に、電源を示す緑 LED と D13 と並列である黄 LED を残し、デジタル出力のテストプログラムがそのまま動作させられるようにした。なお、Arduino は一般的な 5V 動作とするため、Raspberry Pi の GPIO 端子から 5V をとっているが、GPIO の UART 端子は 3.3V であるため、直結すると ARM SoC を破壊する。そのため、Arduino に対して送信する TX (Arduino 側の RX) は直結するが (3.3V であっても ATmega328P は立ち上がり、立ち下がり認識する)、Arduino 側から信号を受け取る RX (Arduino 側の TX) は 5.1K  $\Omega$  と 10K  $\Omega$  の抵抗で分圧し、約 2/3 に降圧した状態

(約 3.3V) で接続している。

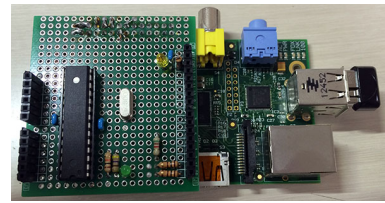


図 1 製作した拡張基板を実装した Raspberry Pi

Arduino と同じように、ブレッドボードを用いて、電源端子と D13 ピンにタクトスイッチを接続することができる。図 2 がその様子である。

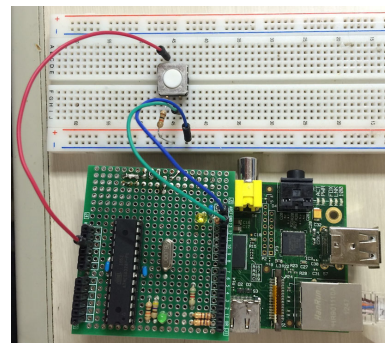


図 2 ブレッドボード上のタクトスイッチを接続

### 3.4 Linux の各種設定

Raspberry Pi の OS は、一般的に利用されている Raspberry Pi 用 Debian である「Raspbian」を採用した。このことで、市販されている Raspberry Pi 各種解説書と同様の操作が可能であるため、自宅に持ち帰って、それら書籍を読み学習を進めることができる。特に、日本語化済みの「Scratch」が同梱されているため、将来、教員となった後、プログラムは Scratch を用いる方針が示された場合にも、インストールすることなく、すぐに試すことができる。

一方、「ドリトル」は同梱されていないため、配布する SD カードのマスターイメージに加えておく必要があった。ドリトルは Java で実装されているため、Raspberry Pi 用に Oracle が配布している Java ランタイムにより、そのまま動作する。2014 年 6 月以後の Raspbian には Oracle Java SE 8 が入っているため、特に Java ランタイムをインストールする必要はなく、かわりにシリアル通信に用いる Java モジュール「RXTX」の CPU 依存部分を ARM 用に置き換える必要があった。また、GPIO に出ている UART 端子 (デバイス名 “/dev/ttyAMA0”) はシリアルコンソールの設定がされているため、getty プロセスの対象からこのデバイスを除くことと、Raspbian 起動設定でシリアルコンソールが指定されているので、その記述を削除する必要があった。Raspbian のウィンドウマネージャは LXDE



が採用されているが、ドリトルがデスクトップアイコンから起動するように、またスタートメニュー相当のメニューにも現れるよう、LXDE の設定を行う必要があった。

ドリトルは日本語命令と日本語識別子でプログラムを記述することができることが特徴であり、英語を極端に忌避する受講生たちに評判がよい。しかし、Raspbian では日本語入力システムは追加インストールする必要がある。Raspberry Pi の RAM は 512MB であり、Java ランタイムは小さいが、X と同時に動作させる必要があるためメモリにあまり余裕はない。したがって、日本語入力システムにもメモリ消費の少ないものが望ましいが、いくつか試してみたものの、操作の違いや辞書サイズの問題から、Windows しか知らない受講生たちにとって大きなストレスとなることが強く懸念されたため、Google 日本語入力のオープンソース版である mozc をセルフコンパイルし、同梱することとした。mozc のプロセスは非常に大きいため、これを起動するだけで LXDE デスクトップ環境すべての動作が非常に遅くなるが、「1 文字 1 秒で入力するしかない。そういうものだ」と授業で説明することで、「遅い」という苦情は最初の 1 回だけで、以後特に問題とはならなかった。

### 3.5 Windows からのリモートデスクトップ利用

実習を行う教室は高等学校と同じ大きさの机、椅子が配置された普通教室であり、Wi-Fi 環境は設置したものの、有線のハブ等は長年使用されず、故障したままのものもあり、普通教室に一時的に展開するような使い方であったため、床が工事されているわけでもない。また、Wi-Fi が当たり前の世代である受講生に、Raspberry Pi のためだけに有線 LAN を強要するのは適切ではないと考えた。また、普通教室であるため、Raspberry Pi に USB キーボードとマウス、HDMI 入力のあるディスプレイを受講生分提供することもできない。そのため、Raspbian の設定は Wi-Fi を前提とし、また、TightVNC Server をインストールすることで、受講生のノート PC から VNC クライアントでリモートデスクトップ接続するという、「Wireless and Headless」な構成を選択せざるを得なかった。

通常 Wi-Fi は WPA2 による暗号を設定するので、Wi-Fi の設定を受講生自身が行うためにはローカルなディスプレイが必要だが、それができない。また、VNC 接続するためには IP アドレス固定にするか、mDNS による広報を用いて接続する Raspberry Pi を見つけ出し（Windows ではエクスプローラーに名前が並ぶ）、その名前で VNC クライアントから接続することになるが、すべてが同じ名称では区別することができないし、IP アドレス固定の場合でも、個別の IP アドレスが必要となる。したがって、授業開始前に SD カードに教室の Wi-Fi の WPA2 接続設定と IP アドレスの設定を個別に行っておく必要が生じた。これに関

する問題は後述する。

## 4. 試行授業実践の結果

準備を入念に行ったことで、大きな問題は発生せず、都度生じた問題については次回までに解決できる範囲に収めることができた。

受講生は、前半のローカル PC で実行したプログラムが、Windows の画面から、無線接続された小さなボードのなかで動いているドリトルでも、非常に低速ながら全く同じように動作することに、安心して 9 回目以後の内容に取り組む様子が観察された。VNC クライアントの制限で Windows からのカットアンドペーストはできないため、手動で打ち直す必要はあっても、同じバージョンのドリトルが同じ見栄えで動くことがよかったというコメントを述べる者もいた。

また、受講生に強い印象を与えたのは、単なる LED 点灯であっても、ケーブルのない、隣の机の上の基板で動いていることで、データやプログラムが基板のなかで処理されていることが歴然とわかること、また VNC サーバが複数クライアントでの同時接続を許すことから、VNC のパスワードは共通なので、他の受講生の画面にログインし、何名かで相談しながら疑問を解消したり、あるいはプロジェクタに接続した PC からお手本となる学生の画面にログインして、これまで生徒として PC 教室で慣れていた、画面共有と同じような、自席から移動せずに発表するような授業展開ができることであった。このことで、残りの自由課題のプロジェクトでは、前年度までの授業では見られなかった、データ加工を明確に意識した、CdS を用いて得られた照度の値を、RGB それぞれ別の関数で加工し、表示はドリトルの画面上ではあったが、色の変化を演出することで、明暗に一定の意味付けを行う例があった（図 3）。

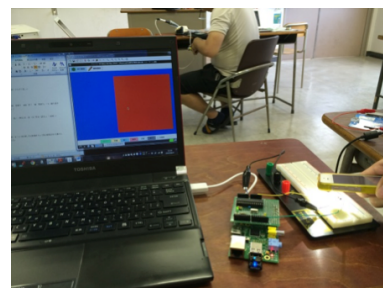


図 3 CdS による明るさから色の変化を伴う照明を作成

これはすぐにフルカラーの LED の制御に置き換えることが可能であり、近年商品化されている、色を演出する照明と同等のものとして実演することもできる。複数の CdS を使わせるよう促すなどで、芸術学校で制作されるインスタレーション作品のような展開も期待できるであろうし、そのことで、コンピュータを通した計測値は抽象的な値として扱われる、（したがって、別の目的に転用することでも

きる)という本質的理解を促進させることにつながる  
ことが強く期待できる。

## 5. 議論

今回の試行授業実践では、学習者の学習動機付けのために、ノート PC から Raspberry Pi 上で動作するドリトルから、限定版 Arduino を経由してセンサとアクチュエータとの入出力を行う環境を準備した。学習者はローカル PC と同じ操作により、無線接続された Raspberry Pi 上の環境でプログラム作成を行うことができることから、新鮮さを覚えつつも、戸惑いなく準備した実習内容をこなすことができたという意味では、当初の目的は達成できたと考えている。また、最終プロジェクトで、一部に自主的にデータを加工して、付加価値をつけた結果を表示する例も見られたことから、物理的なケーブルを廃しながらも、PC のリモートデスクトップを通して動作を把握し続けられることで、データ処理をイメージしやすくなるのではないか、という仮説を得ることができたと考える。

この意味は、一般的な自律型ロボットカーのプログラムでは、ロボットカーが取得している値がどのようなもので、プログラムのどの位置を通っているかを確認することが外見からはわからないため、ロボットカーの動作であればモータ回転時間の調整を試行錯誤するなど、合理的でないが取り組みやすい対処に陥ることが多いのに対し、ロボットシミュレータでの試験を通して解決をはかる [5] 手段とは別のアプローチとなりうると考えられる。

一方で、学習環境の設計に起因する課題として、Wi-Fi の設定が自由にできないことが挙げられる。これは、自宅に持ち帰った後、継続してこれを利用するためには自宅 Wi-Fi の設定を何らかの手段で行わなければならないが、ノート PC とスマートフォンで IT に関わる生活のすべてが完結している現代の大学生にとって、USB 接続のキーボードやマウスの調達と自宅 TV への接続の必要性は非常に敷居が高い、という意味で「できない」という苦情を多く寄せられたことで気づいた問題である。自室用 TV でさえ、現代の大学生はワンセグもしくはネット視聴で足りるため持たないということである。

現在の基本的設計を維持したままこれを解決するためには、他の通信チャンネル、例えば Bluetooth 経由での操作を加えるか、何らかのディスプレイと簡易な入力装置を拡張器版として追加することが考えられるほか、一般の Wi-Fi 機器同様の、ルータのボタンによる自動設定に対応する方法も考えられる。これらについて、情報収集を行い、何らかの手立てを講じる必要がある。

## 6. おわりに

計測と制御を行うプログラムを学習させるための環境として、PC の画面でのプログラム学習をそのまま無線接続

されたマイコン基板に実現するため、Linux が動作するマイコン基板「Raspberry Pi」とアナログ入力を含め基板の I/O に直接触れないようにするため簡易 Arduino を搭載した拡張器版の組み合わせを実現し、15 名の学習者を対象に授業を行い、一定の成果を得た。しかし、「Wireless and Headless」という厳しい条件は、授業を行う教室環境に依存した設定に頼ることから、そのままでは自宅学習に使用できないという問題が残った。今後、問題点の解決を図るとともに、今回得られた成果をより情報技術の本質的な学習を高めるためのカリキュラムの改善を行う予定である。

## 参考文献

- [1] 鎌田敏之: 教員養成におけるロボットを用いた計測制御の授業実践, 研究報告コンピュータと教育 (CE), 2009-CE-99(11), 1-6 (2009).
- [2] 兼宗進, 久野靖: プログラミング言語ドリトル: グラフィックスから計測・制御まで. イーテキスト研究所, 第 2 版 (2011).
- [3] 文部科学省: 中学校学習指導要領解説 技術・家庭編 (2008).
- [4] Department of Education, UK: *National Curriculum in England: computing programmes of study* (2013).
- [5] 和田佑介, 大西修平, 紅林秀治, 井戸坂幸男, 兼宗進: 教育用制御ロボットの動作シミュレーション機能の実現, 情報処理学会第 74 回全国大会講演論文集, pp. 649-651 (2012).