

mruby を使ったプログラミング教育教材の開発

九州工業大学 情報工学研究院 田中 和明 (Kazuaki Tanaka)
Faculty of Computer Science and Systems Engineering
Kyushu Institute of Technology

1 はじめに

2020年に小学校からプログラミング教育必修化されることとなった。プログラミングをどのように教育するかという方法論について議論が進められてきている。同時に、プログラミング教育において、何を使って教えるのかという点も議論が必要であろう。

筆者は、公的機関のサポートのもと、実践的なプログラミングの演習を数年に渡って行っている。演習は、実際に小中学生が自らのアイデアをプログラムにより具体化させる内容である。この演習で使用する教材は、いかに簡単かつ短時間でプログラムを動かすことができるか、という点を重視して開発されている。

本稿では、プログラム言語 mruby を使った教材の概要と特徴、教材を利用して実施した演習の報告をする。

2 IoT教育

2.1 Internet of Things

IoT（モノのインターネット）とは、さまざまなモノをネットワークと接続することで、環境情報を収集し、サービスとして利活用する仕組みである。IoTでは、さまざまな環境、さまざまなモノ（デバイス）、さまざまなネットワークが関係する。これらIoTに関係する技術範囲は広く、IoTを総合的なプログラミング教育に利用することができる考えた。

ここで、IoTの教育を考えると、大きくセンサデバイスなどのモノに近い領域と、ネットワークとサービスへの利活用というインターネット（クラウド）に近い領域に分けることができるだろう。

センサデバイスに近い技術は、ハードウェアを直接的に扱い、デバイスの動作を実体験できることから、IoTを学ぶ入り口としてわかりやすい。そこで、筆者は、この技術領域を教育教材として提供することを試みている。

2.2 IoT教育で得られるもの

IoTを通じて何かを教育しようとする時、IoTが利用する技術に着目する。多くのIoTシステム¹では、センサデバイスから環境情報を取得し、そのデータを処理して、クラウド等のネットワークに送信、その後、データ解析を行うモデルである。図1のように、ハードウェアとソフトウェアが相互に連携した構成となり、IoTシステムを作ろうとすると、ハードとソフトの両方の知識が必要となる。

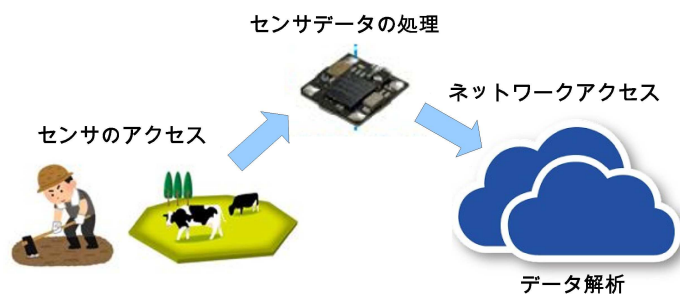


図 1: 典型的な IoT システム

IoTシステムに要求される要素技術を以下に示す。

センサのアクセス ハードウェアへの直接的なアクセスが必要で、アナログ信号、デジタル信号の扱い、簡単な電気回路の技術が含まれる。

センサデータの処理 センサデバイスで得られた生のデータに対して、物理量との差異を補正する演算、データ圧縮などの前処理が含まれる。

ネットワークアクセス クラウドサーバとの接続など、ネットワーク技術が含まれる。

データ解析 クラウドに集約されたセンサデータに対する、集計、解析、可視化などの技術が含まれる。

このうち、初等教育におけるプログラミングでは、センサのアクセスとセンサデータの処理を対象とする。ネットワークアクセスは、ネットワークプロトコルの理解が必要であり、内容が専門的であることと、データ解析には高校数学以上の知識が必要になるため、これらを除外した範囲としている。一方で、データの取得だけでは、プログラミングのモチベーションを維持することはできないため、何らかのフィードバックを持たせるようにする。

¹通常、IoTシステムといった表現は使わない。ここでは、あえてハードとソフトの組み合わせで有ることを意図して「システム」と呼んでいる。

2.3 IoT 教育カリキュラム例

小中学生向けのIoT教育教材と教材を使ったカリキュラムを構成する。カリキュラムは、以下の技術を対象とする。

- センサのアクセス。ハードウェアへの直接的なアクセスの技術と、簡単な電子回路を対象とする。
- センサデータの処理。センサから得られたデータ（センサ値）について、簡単な条件判定を伴う制御プログラムを対象とする。
- ユーザへのフィードバック。簡単なユーザインターフェースとして、LED もしくはLCD を利用した出力を対象とする。

以上の内容を含むカリキュラムとして、以下の全5項目を構成した。例えば1つの項目を2時間で実施として、全体で10時間の実施を想定している。

1. プログラミング基礎。後述するマイコンボードのみを使い、10行以下の簡単なプログラムを通じて、プログラムの動作の流れを理解する。
2. 電子回路の作成。プログラムにより、デジタル信号の出力を行い、その出力を利用してLEDを点滅させる。ソフトウェアとハードウェアの関連について理解する。
3. プログラミング応用。入力インターフェース（電氣的なスイッチ等）を使い、入力状態の条件判定による分岐を伴うプログラムを使い、複雑なプログラムの動作を理解する。
4. センサの利用。環境センサ（例えば温度センサ）を使い、環境の変化に対応してプログラムの実行を変化させ、ユーザに提示するプログラムを作成する。これにより、IoTの基礎であるセンサ入力とその応答を理解する。
5. 創作活動。自分自身で、実現したいふるまいを考え、そのアイデアをプログラムとして具体化させ、実際のハードウェアの動きを検証する。自由なアイデアを試行錯誤を経て動くモノとして完成させる体験得る。

3 mruby を利用した教育教材

教育的教材として実現可能な内容とするため、教材を使うために必要となる開発環境（PC等）や必要な機材（電子部品等）は入手性の高いものに限る。また、コストも可能な限り低減させる。この教材の選定と必要な環境の開発について、説明する。

3.1 マイコンボードの選定

提案する教材では、簡単な電子回路を構成し、センサからのデータの取得、プログラムによるデータ処理、ユーザへの出力を行う。これらの動作を実現するために、適切な入出力機能を持ち、ある程度のプログラムが実行可能なマイクロコントローラ（以下、マイコン）を選定する。さらに、作成したプログラムをマイコンに流し込む機能も必要である。これらの要件と、入手性の高さ、価格を鑑み、Cypress Semiconductor 社（米国）製の PSoC5LP マイコンを採用した（図2）。

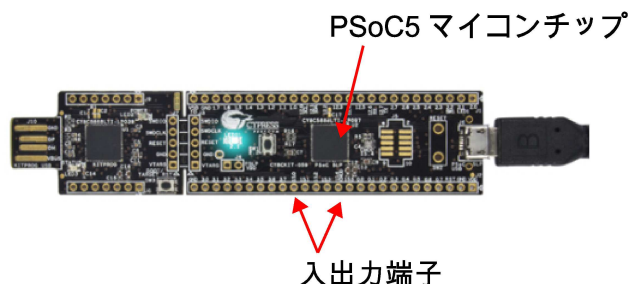


図 2: PSoC5 マイコン

このマイコンには、単純なデジタル・アナログの入出力のほか、プログラム可能なデジタル論理回路、アナログ回路、信号生成回路（PWM など）、通信インターフェース回路（I2C、SPI、Serial など）もチップ内に内蔵しており、必要な電子回路を外部に持つことなく、単一のチップで完結できる。市場価格は、10 ドル（2017 年時点）であり、容易に入手できる状態にある。

3.2 Ruby, mruby

プログラミングに慣れていない初学者向けに IoT 教育教材を提供する。ソフトウェア部分については、プログラム言語 Ruby を利用した。

通常、IoT ソフトウェアの開発には C 言語を用いる。しかし、C 言語の習得には、多くの時間を必要とする。例えば、筆者の所属する九州工業大学では、C 言語によるプログラム開発の習得に大学 1 年生の前期・後期を費やしており、総演習時間は 90 時間に及ぶ。

プログラミング教育の本質は、プログラム言語の習得ではなく、プログラムを利用した問題解決であると考えており、プログラムの実行性能の高さよりも、開発しやすさを有線すべきであろう。そこで、可読性が高いことで知られる、プログラム言語 Ruby を採用した。

図3に同じ実行結果を得るための C 言語と Ruby のコードの比較を示している。おそらく、多くの読者にとって、Ruby のコードを読めばその挙動を把握できるのではないか

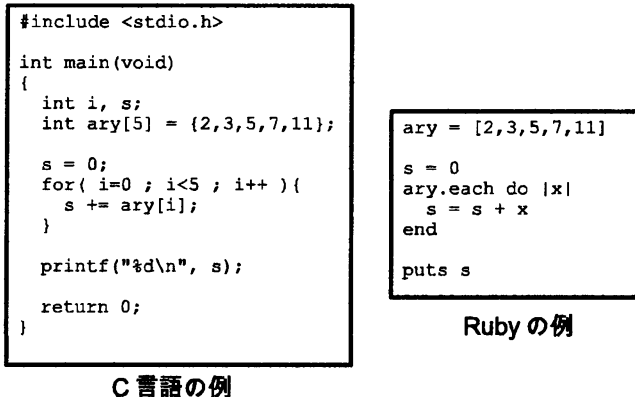


図 3: プログラムコードの比較

と思う。また、プログラムの各行は、ほぼ一対一に対応しており、プログラムのロジックは全く同じであることも分かる。

プログラムは小型マイコンを使って動作させる。このような実行環境においては、プログラムが実行時に使用できるメモリ（RAM）に制限がある。採用したマイコンの場合、実行時には 64KB のメモリのみが使用できる。使用できるメモリの中から、開発環境からのソフトウェアの転送やマイコン起動時に必要な初期化動作などで必要なメモリも確保する必要があるため、実際にプログラムが使用できるメモリは少ない。

一方で、プログラム言語 Ruby は、インタープリタ型言語であるため、実行にソースコードを解析しながら逐次実行する。このソースコードの解析には多くのメモリを必要とするため、そのままの方法では Ruby を舞い込んで動作させることはできない。

筆者は、過去の研究開発において、Ruby を軽量化したプログラム実行環境 mruby を開発した²。この成果物を、本教材に採用することで、使用できるメモリに制約がある環境下での Ruby プログラムの実行を実現した。

3.3 開発環境

教育カリキュラムを実現するため、以下のハードウェア機能を開発した。図 4 に、マイコン内の入出力モジュールを示す。

デジタル入力 ボタンスイッチによる入力状態を得るための入力機能。

デジタル出力 LED への出力を想定した信号出力機能。内部に定電圧源をもたせることで、抵抗器による電流抑制の回路を不要とした。

アナログ入力 AD 変換器によるアナログ電圧の入力機能。

²経済産業省 地域イノベーション創出研究開発事業による軽量 Ruby の開発

アナログ出力 PWM によるアナログ電圧の出力機能。

ウェイト機能 プロセッサを一定時間停止させることによる時間待ち機能。

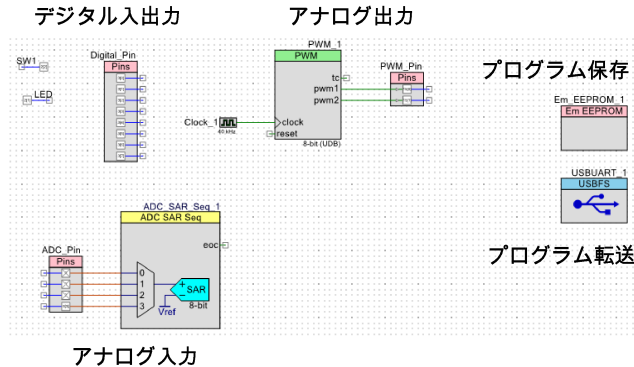


図 4: ハードウェア機能

このハードウェアに、mruby プログラムを書き込んでプログラムを実行させる。mruby プログラムの作成（エディタ）、コンパイル、マイコンへの転送を提供するために、統合開発環境を開発した。

プログラムをエディタ部に記述し、メニューを選択するだけで、コンパイルとコンパイル済みのオブジェクトの転送を行う。この統合開発環境は、マルチプラットフォームアプリケーションとして作成されており、Windows10、MacOS、Linux においては、インストールや USB ドライバ等のインストールを行わずに実行できる（Windows 10 以前の Windows OS においては USB ドライバのインストールが必要となる）。

統合開発環境は、オープンソースソフトウェアとして公開しており、共同研究を行っている、しまねソフト研究開発センター（島根県）からダウンロードできる³。

統合開発環境を利用している様子を図 5 に示す。

この統合開発環境により記述された Ruby のプログラムは、コンパイルによりバイトコードに変換される。mruby の特徴は、プログラムをコンパイルした後であっても、デバイス非依存のバイナリコード（バイトコード）を生成する点にある。この仕組みにより、コンパイル済みのプログラムを、異なる実行環境（マイコン）で動作させることができる。

本稿では、PSoC5LP マイコンを使用した教材を紹介しているが、他に Nordic 社（ノルウェー）の nRF52、STMicroelectronics 社（スイス）の STM32 マイコン、Microchip Technology 社（米国）の PIC16F にも対応しており、全く同じコンパイル済みバイナリコードを動作させることができる。また、汎用 OS である Windows、MacOS、Linux にも対応している。

³<http://www.s-itoc.jp/activity/research/mruby/>

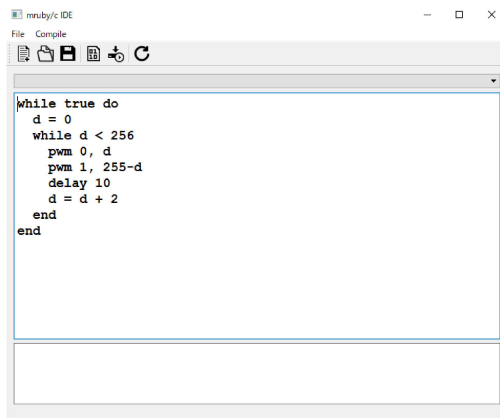


図 5: 統合開発環境によるプログラム作成

多くの実行環境に対応する際に問題となるのが、ハードウェアの差異を吸収する仕組みである。mrubyでは、mrubyVM（mruby 仮想計算機）とHAL（ハードウェア抽象化）により、ハードウェアの差異を吸収し、異なる実行環境で同じプログラムの動作を実現している。

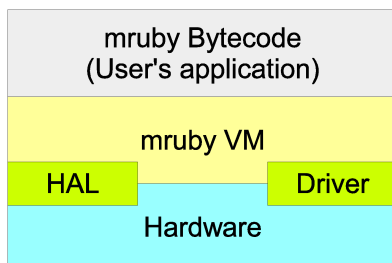


図 6: mrubyVM

mrubyVM そのものは、HAL のハードウェア抽象化を司るため、ハードウェアに依存する。しかし、ある実行環境向けの mrubyVM を一度だけ開発（他の実行環境の VM をポーティング）する必要があるが、VM さえ開発してしまえば、mruby のプログラム実行環境を得ることができ、それ以降のソフトウェア開発が大幅に簡単化される。結果として、開発者は実行環境の違いを意識すること無く、プログラムを開発できるようになる。

4 事例紹介

本稿で紹介した教育教材を利用して、小中学生向けの複数の演習を実施した。これらの一部を紹介する。

4.1 事例 1: 島根県

島根県商工労働部産業振興課の主催で、平成 26 年度から毎年、島根県内において演習「最先端 IT キャンプ in SHIMANE」を実施した。直近に実施した平成 29 年 8 月の島根県益田市での開催について紹介する。

益田市内の宿泊施設で一泊二日の合宿形式で実施し、小学 4 年生から中学 2 年生までの市内の小中学生 16 名が参加した（図 7⁴）。

小学生はキーボードに慣れていないこともあり、最初はプログラムの入力に時間がかかっていたが、子どもたちがキーボードやプログラミングを習得する速度は早く、合宿二日目になるとスムーズに演習が行えていた。

中学生と小学生の混成グループをつくり、互いに教え合うような環境を準備した点も、演習がスムーズに進んだ理由の一つであると考えている。上級生にとっては、下級生に教えるということが良い経験となり、下級生も気軽に聞くことができる人が近くにいることで、「何をすればよいかわからない」という事態を回避できた。



図 7: 島根県益田市における演習

⁴利用目的を限定して写真利用の許諾を得て掲載しているため、転載禁止

4.2 事例2: 米国サニーベール市

Sunnyvale Sister City Association(カリフォルニア州サニーベール市)の主催で、平成27年度から毎年、サニーベール市において演習を実施している。直近に実施した平成29年3月の開催について紹介する。

サニーベール市内の市コミュニティセンターを利用し、二日間での演習を実施し、奨学3年生から中学2年生までの市内の小中学生12名が参加した(図8⁵)。



図 8: サニーベール市における演習

サニーベール市はシリコンバレーの中心に位置し、小中学生のITスキルも高い。すでに何らかのプログラム言語を知っている参加者も多く、プログラム作成での問題はほとんど発生しなかった。一方で、簡単な電気回路を含むハードウェアを扱ったことがある参加者はおらず、LEDやジャンパー線などの部品であっても彼らにとっては新鮮なものであると感じられた。また、自分自身が作成したプログラムで、モノが動くことがプログラミングの適用範囲の広さを感じさせるものであったようで、多くの参加者がさまざまな拡張を試みて、予定の時間を越えて熱中していた。

5 今後の展開

本稿で紹介した教材を使った演習は、これまで数回に渡って実施してきており、演習を実施した小中学生の延べ人数は120人を超える。この試行を通じて、教材の内容を充実させてくるとともに、誤った操作による障害を発生させないような仕組みも導入してきた。

例えば、演習において最も多いトラブルは、電気回路の短絡(ショート)である。実行環境であるマイコンと電子部品をジャンパー線などで結合して回路を構成するが、使用する部品数が多くなると、意図しないジャンパー線の接触などによる短絡が発生する。このような想定外の事態についても、マイコンに自動シャットダウン機能を持たせることと、定電流源による出力を行うことで過電流を防止する仕組みを導入した。

⁵利用目的を限定して写真利用の許諾を得て掲載しているため、転載禁止

今後も、各公的機関の協力を得て、継続的に演習を実施していく予定である。ある機関からは、環境センサを用いた複雑な動作の演習の希望が上がっており、近日中にカリキュラム設定と実施の計画を進める予定である。

筆者と関係機関による取り組みが、小中学生のプログラミング教育の効果的な実施に寄与することを願っている。

謝辞

本研究は、京都大学数理解析研究所共同事業「数学ソフトウェアとその効果的教育利用に関する研究」による成果である。