

ビジュアルプログラミング環境「Tsumicky」のご紹介

Introducing the visual programming environment "Tsumicky"

藤本 壱

群馬医療福祉大学医療技術学部

プログラミングを初めて学ぶ人にとって、ブロックを組み合わせるプログラミングを行うビジュアルプログラミングは、比較的なじみやすいものである。そのようなプログラミング環境としてはScratchが多用されているが、Scratch以外にも同様のプログラミング環境は複数存在している。

本稿では、発表者である藤本 壱が開発を進めている「Tsumicky」というビジュアルプログラミング環境を紹介する。GoogleのBlocklyをベースとした環境であり、多様な用途に使えるようにすることを目指して開発を進めている。

キーワード：ビジュアルプログラミング、電子工作、データ分析

1. はじめに

プログラミングを初めて学ぶ方のために、Scratchなどのビジュアルプログラミング環境が多用されている。そのような中で、発表者である藤本壱も、Googleが公開している「Blockly」をベースとして、Scratchと似たビジュアルプログラミング環境である「Tsumicky」を開発している。今回の発表では、このTsumickyの概要をご紹介します。

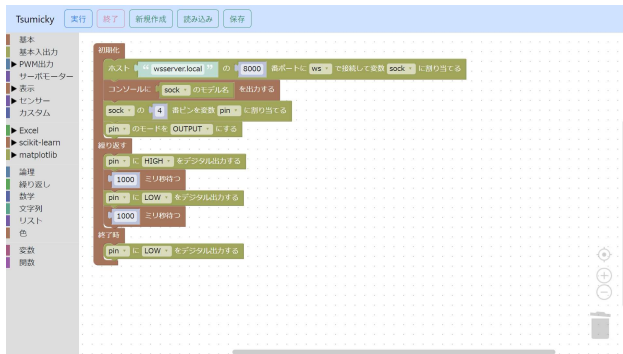


図 1：Tsumicky

2. 開発の動機

発表者は、所属先の群馬医療福祉大学で、「プログラミングの基礎」という授業を担当しており、Pythonを教えている。しかし、プログラミング経験がない学生にとって、Pythonでのプログラミングは簡単ではない。中でも、命令のスペルを間違えたり、

かっこなどの記号を間違えたりといった文法エラーによってプログラムが動作しないことが発生し、そこでつかえてしまう学生が少なくない。

Pythonに限らず、英単語や記号を組み合わせる記述する言語では、前述したような問題を避けることは困難である。そこで、ビジュアルプログラミング環境を利用することで、このような問題を解決することができるのではないかと考えた。

ビジュアルプログラミング環境の世界では、米マサチューセッツ工科大学が開発したScratchがデファクトスタンダードである。Scratchには拡張機能という仕組みがあり、拡張機能を開発すれば、Scratchに機能を追加することが可能である。

ただ、拡張機能を開発する形だと、Scratchに縛られてしまう面があり、自由度があまり高くない。そこで、Googleが提供している「Blockly」をベースにして、ビジュアルプログラミング環境を開発することにした。ブロックを組み合わせるといったプログラミングスタイルにちなんで、「積み木」をもじって「Tsumicky」と命名した。

Blocklyは、「ブロックを組み合わせるプログラムを作る」というユーザーインターフェースや、ブロックの組み合わせを各種の言語（Python、JavaScriptなど）に変換する機能を提供するライブラリである。Scratch風のビジュアルプログラミング環境を手軽に構築することができ、なおかつ自由度が高いため、Blocklyをベースにすることにした。

3. Tsumicky の機能

本稿執筆時点のTsumickyには、以下で述べるような機能が存在する。

3.1. 電子工作向けプログラミング

Tsumickyの開発を着想した時点では、電子工作向けのビジュアルプログラミング環境を作りたいことを想定していた。そのため、現時点でのTsumickyの機能の中心となるのは、電子工作向けのプログラミングである。

Scratchにも一部のマイコン向けのプログラミングを行う機能があるが、それと似たような仕組みを取っている。マイコン側には、パソコンからの通信に反応して処理を行うためのファームウェアを書き込んでおく。そして、パソコン側でプログラムを実行すると、パソコンとマイコンの間で通信が行われ、処理が行われるという形で動作する。

本稿執筆時点では、以下のマイコンに対応している。

- ESP32(Espressif Systems)
- ESP32-S3(Espressif Systems)
- ESP32-C3(Espressif Systems)
- Raspberry Pi(Raspberry Pi Foundation)
- Raspberry Pi Pico(Raspberry Pi Foundation)
- Raspberry Pi Pico W(Raspberry Pi Foundation)

また、以下のような機能を持っている。

- 基本的な入出力（デジタル入出力、PWM出力、アナログ入力）
- サーボモーターの制御
- 温度／湿度／気圧センサーの制御（DHT11／DHT22／BMP280）
- モーションセンサーの制御（MPU6050）
- キャラクタ液晶ディスプレイへの表示
- NeoPixel(フルカラーLED)の制御

図2はプログラムの一例で、BMP280から得た温度と気圧の情報を、キャラクタ液晶ディスプレイに表示し続けるものである。実行したときの結果は図3の通りである。

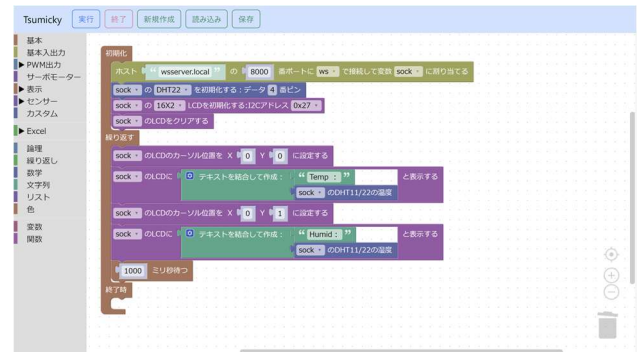


図 2：温度と気圧の情報を表示するプログラム

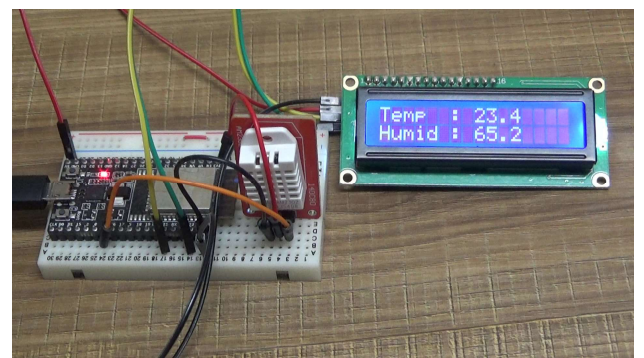


図 3：図 2 の実行結果

3.2. Excel の操作

当初は電子工作向けのプログラミング環境を作りたいことを目指していたが、開発を進める中で、「多様な用途に使えるようにすれば、Tsumickyの可能性がより広がるのではないか」と思うようになった。

そのような中で、Excelを操作することを思い立った。Excelにはマクロ機能があり、定型的な処理を行うことが可能である。ただ、そのマクロ言語はBasic(Visual Basic for Applications)であり、決して簡単とは言えない。Excelをビジュアルプログラミングで操作できるようになれば、需要があるのではないかと考えた。

また、電子工作をする中で、センサー等から得たデータを記録したい場面はよくある。この用途のためにも、Excelとの連携は役に立つと考えた。

Excelは多機能であり、そのすべてに対応するには非常に時間がかかる。そのため、Tsumickyでは以下のようなごく一部の機能のみ対応している。

- ファイルの読み書き
- シートの操作（シートの作成など）
- セルの操作（値の読み書き、書式の設定など）

図4は、Excelを操作する例である。A1~E5セルに1~25の数を順に入力し、なおかつ背景が市松模様になるようにしている。

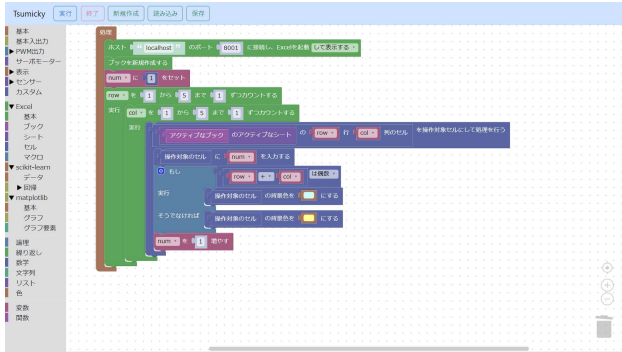


図4：セルへの数値の入力と書式の設定の例



図5：図4の実行例

3.3.scikit-learn/matplotlibを使ったデータ分析

ここ数年、データサイエンスの分野が人気となっている。統計、機械学習、プログラミングなどを組み合わせて、大量のデータを分析して知見を得ようというような学問である。

データサイエンスを学ぶ上で、プログラミングは必須と言える。しかし、ここまで述べてきたように、プログラミングは決して簡単ではない。そこで、データサイエンスの方面にもTsumickyを生かすことができれば良いのではないかと考えた。

その手始めとして、Pythonのデータ分析ライブラリとして有名な「scikit-learn」と、グラフ描画ライブラリの「matplotlib」を、Tsumickyで操作する仕組みの開発に着手した。本稿執筆時点ではまだ実験的な段階で、回帰分析を行う機能のみ実装した。

Excelのワークシートに用意したデータを元に回

帰分析を行うことができ、またその結果をmatplotlibでグラフ化して表示することができる。

図6は、Excelのデータを元に単回帰分析を行い、また散布図を表示する例である。また、その結果は図7のようになる。

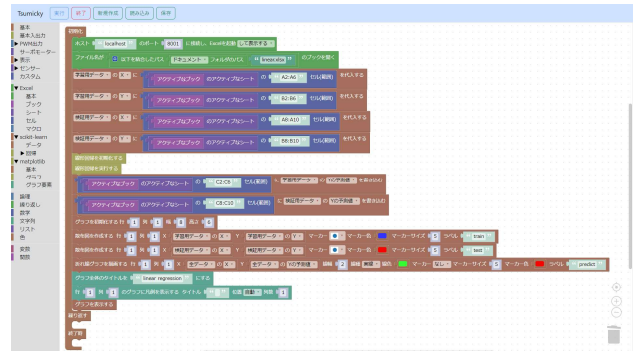


図4：セルへの数値の入力と書式の設定の例

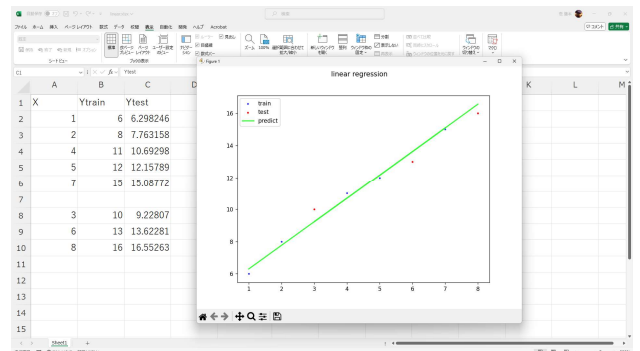


図5：図4の実行結果

4. 基本的な動作の仕組み

TsumickyはWebブラウザ上で動作する。作成したプログラムは、Blocklyの機能によってJavaScriptに変換され、Webブラウザ上で実行される。そして、そのプログラムと、外部（マイコン等）との間で通信を行い、外部のものを制御するという仕組みになっている。

外部（マイコン等）とはWebSocketで通信を行っている。WebSocketは、通信の開始はWebブラウザ側から行うが、その後はWebブラウザと通信相手との間で双方向に通信することができる仕組みである。

現状では、Tsumickyから外部への一方の通信しか行っていないが、外部の状況の変化（例：マイコンに接続したスイッチが押された場合）に応じて処理を行いやすくするために、外部からTsumickyへの通

信を行うことを予定している。そのためにWebSocketを利用した。

Excelの操作については、Excel自体にはWebSocketで通信する機能がないため、通信の仲立ちをするサーバーをPythonで作成した。そして、そのサーバーに、scikit-learnやmatplotlibとの連携の機能も持たせるようにしている。

また、Tsumickyと外部との間でやり取りするデータは、JSON(JavaScript Object Notation)の形式にしている。JSONには、「g」(グループ)／「sg」(サブグループ)／「c」(コマンド)／「p」(パラメータ)のキーが含まれる。g/sg/cで実行する処理を特定し、また処理を行う際に必要なパラメータをpで渡す。

例えば、マイコンの5番ピンの値をHIGHに設定する場合、Tumickyからマイコンに対して、以下のようなJSONを送信する。

```
{
  "g": "bio",
  "c": "dw",
  "p": {
    "p": 5,
    "v": 1
  }
}
```

5. 今後の予定

Tsumickyの機能は、現状ではここまで述べた通りであり、まだ機能的に十分とは言えない状況である。今後も開発を進めて、多様なプログラミングを手軽に行えるようにしていきたいと考えている。

例えば、音声認識と連携することで、声によって指示を出せるようプログラムを作ることが可能になる。音声認識だけでなく、画像認識などのAI関係との連携は、ぜひ追加したいと考えている。

また、Tsumickyは現状でGitHubで公開している（ただし、データサイエンス系の機能は本稿執筆時点では未公開）。アドレスは以下の通りである。

<https://github.com/hajimef/tsumicky>

多くの方にTsumickyを使っていただき、フィードバックを得て、さらに発展させてゆきたいと考えている。