

マイコン用ビジュアルプログラミング言語 ArduBlock の メニュー構成等の検討

光永 法明 (大阪教育大学)

1 はじめに

ビジュアルプログラミング言語を使うことで、キーボード操作に不慣れな児童・生徒も、プログラミングを楽しむことができる[1]。マイコン/マイコンボードならびに組み込み向けのビジュアルプログラミング言語と開発環境としては、レゴマインドストームをはじめ、多くの教材会社やキットメーカーが作成したものがある。しかし、回路構成の自由度が高く、入手しやすい開発環境は、それほど多くない。

そのような環境としては Logicator for PICAXE と ArduBlock があげられる。PICAXE は専用ファームウェアが動作するマイコンで、Logicator を使うとフローチャートのようなブロックと矢印の組み合わせでプログラムを作成できる。ArduBlock を使うとブロックを組み合わせ、Arduino マイコンボードで動作するプログラムを作成できる。また ArduBlock [2] はソースコードを公開した、いわゆるオープンソースで開発が進められており、今後の発展が期待される。

ブロックを組み合わせ、プログラムを作成するプログラミング環境では、言語を構成するブロックを選択、配置してプログラムを作成する。そのため、文字で記述するプログラミング言語の、キーワード (if, for など)、関数や命令 (digitalWrite, OUT など) 等に相当するブロックが多数用意される。多数のブロックの中から目的を達成するブロックを見つけやすくするには、適切なブロックの分類、配置が必要である。

ArduBlock ではブロックをいくつかのグループに分類して、メニューに配置している。しかし、グループの分類 (メニュー構成) をみたとき、初心者向けとは必ずしもいえなかった。そこで、ArduBlock について初心者により分かりやすいメニュー構成を検討したので報告する。

2 メニュー構成の検討

Scratch [3] は子ども向けのプログラミング環境として有名であり、多くの子どもたちが利用している (図 1)。Scratch はプログラミング言語を教えプログラムを作らせることを目的としておらず、子ども自身が興味を持って何かを達成しようとすることを実現するためのツールを目標としている。したがって Scratch では、子ども



図 1: Scratch 2.0 の表示 (一部切り抜き)

のやりたいことでブロックが分類、メニューが作成されている。図 1 にあるように、たとえば、キャラクタを動かしたいときには「動き」を、キャラクタの見た目を変えたいときには「見た目」を、音を出したいときには「音」を、マウスによるクリックなどのイベントに反応させたいときには「イベント」を、繰り返しや条件分岐などプログラムの流れを制御したいときには「制御」をクリックすれば、その分類のブロックが表示される。またブロックの色は、メニューの色と統一されていて、同じ分類のブロックのメニューを探しやすくなっている。

ところで、マイコンを利用した作品で何をしたいかという問をすると、最初の答は光らせたい、音を出したい、動かしたいといった答になると考える。それができてから、具体的に、点滅させたい、音を変化させたい、動きを変えたいとなり、さらには外界に反応させたいとなるのではないだろうか。

一方、最近の ArduBlock (2013 年 10 月 31 日のベータ版) を見るとブロックは「制御」「ピン」「比べる」「計算する」「変数/定数」「ユーティリティー」「通信」などに分類されている。この分類はプログラミング言語の構造を知っていると分かりやすい。たとえば「制御」には for 文, while 文, if 文と時間待ちの、「ピン」には入出力ピン操作の、「比べる」には比較演算子と論理演算子に相当のブロックが分類されている。LED を点灯するには「ピン」に、音を出すには「ユーティリティー」にあるブロックを使う。どちらかという、マイコンにできることを理解してから、必要なブロックを探す必要がある。以下では、ArduBlock のブロックの分類を作品を作る過程を考慮して見直す。

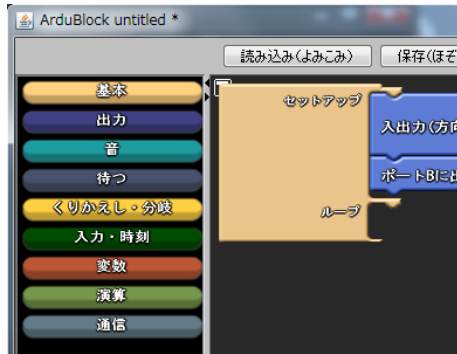


図 2: 変更した ArduBlock のメニュー部分

3 ブロックの分類と実装

LED の点滅, 音の出力, モータの動きといったものがあると, 回路・マイコンの動作が分かりやすい。また作りたい作品のイメージを具体化するにも役立つと考える。自身の経験も含め, 初心者のマイコンプログラム (作品) の作成過程は次のようになると考える。

1. 作品づくりの最初の段階では, 出力に関するブロックの使い方を覚える。
2. つぎに, そういったブロックと時間待ちのブロックを並べ, オープンループで動作するプログラムを作る。
3. 繰り返しや, 条件分岐, あるいは演算を使って動作を複雑にする。
4. 入力に応じて動作を分岐するプログラムにする。
5. 入力や記憶 (変数) を使って計算 (演算) をし, 動作を変えるプログラムにする。
6. デバッグ方法も覚え, さらに発展させていく。

この仮定に基づいてブロックを, 1) プログラムを構成する基本のブロック (基本), 2) デジタル/PWM 出力 (出力), 3) 音の出力 (音), 4) プログラムの待ち (いわゆる busy loop, 待つ), 5) くりかえし・分岐, 6) デジタル入力と時刻 (入力・時刻), 7) 変数, 算術・論理・ビット・比較演算 (演算), 8) 非同期シリアル・I2C 通信 (通信) に分け, 順にメニューに並べ 2012 年 7 月ごろの ArduBlock を基に実装した (図 2)。Scratch にならい, ブロックとメニューの色を統一し, ブロックを探しやすくしている。例外としては, 形状が同じで区別が必要な, 変数, 定数と音の長さをテンポから計算するブロックについて, メニューとは別の色分け (値を代入できる変数はピンク, 定数と計算ブロックは赤色, 2 進数の定数は暗い赤) をしている。



図 3: 追加したブロックを利用したプログラムの例

また ArduBlock では Arduino マイコンボードを前提としているが, ヒダピオシステム [4] で学んだ生徒に使わせたいという要望に合わせ, ATtiny2313 マイコンの利用を考慮した変更も行った (図 3)。Arduino ではピン単位の操作を簡単にできるようにし, マイコンのポートレジスタを意識せずに済むようになっている。一方, ヒダピオシステムではポートレジスタを直接操作する。そこでポートレジスタや入出力を決めるレジスタを操作出来るようにしている。また音の出力について, 音階に応じた周波数を返す定数ブロックと, テンポに応じて音符の長さから時間を計算するブロックを用意している。

4 まとめ

本論文では, 初心者のマイコンプログラムの作成過程を考慮し, ビジュアルプログラミング言語 ArduBlock のブロックの分類を提案して変更を実装した。分類が他のビジュアル言語の参考に, また変更した ArduBlock を活用いただければ幸いである。

謝辞

助言と要望をいただいた兵庫県三田市立狭間中学校の浅田寿展教諭に感謝する。本研究は JSPS 科研費 25870418 の助成を受けたものである。

参考文献

- [1] 兼宗, 阿部, 原田: プログラミングが好きになる言語環境. 情報処理, vol.50, no.10, pp.986–995, 2009.
- [2] ArduBlock: <http://blog.ardublock.com/>
- [3] J. Maloney, M. Resnick, N. Rusk, B. Silverman, and E. Eastmod: The Scratch programming language and environment. Trans. Comput. Educ., vol.10, no.4, pp.16:1–16:15, 2010.
- [4] ヒダピオシステム: <http://hidapio.jp/>