

「関数」の概念を学習要素に導入した タンジブルなプログラミングツール

津田真理子^{*1} 本吉達郎^{*2} 鉄村直樹^{*1} 澤井圭^{*2} 増田寛之^{*2} 玉本拓巳^{*2}
小柳健一^{*2} 大島徹^{*2}

A Tangible Programming Tool Incorporating the Concept of Functions as a Learning Element of Programming

Mariko Tsuda^{*1}, Tatsuo Motoyoshi^{*2}, Naoki Tetsumura^{*1}, Kei Sawai^{*2},
Hiroyuki Masuto^{*2}, Takumi Tamamoto^{*2}, Ken'ichi Koyanagi^{*2} and Toru Oshima^{*2}

Abstract - We are developing P-CUBE system which is a programming tool that users are able to make a program by easy operation. Since, users cannot learn a concept of “Function” using P-CUBE, we propose P-CUBE2 adding functions of learning its concept. We introduced a function mat, function blocks and hiragana blocks into P-CUBE2. An experiment for comparing task that is using function blocks and not using function blocks was conducted to verify whether subjects can learn the benefits of the function using P-CUBE2. This result suggested that users can learn the advantage of the use of a function using P-CUBE2. This paper describes system of the P-CUBE2 which is added function as extension of learning contents and experiment using that tool.

Keywords: Programming tool, Function of programming and Tangible interface

1. はじめに

プログラミング教育では、パーソナルコンピュータ（以下、PC）にインストールされたプログラミングソフトを用いることが一般的である。しかし、PCを操作する過程で学習者がキーボード操作やソフトの扱い方の習得に集中してしまうことが懸念される。このことから、文法習得の段階で躓き、プログラミングに対する苦手意識の形成やモチベーションの低下につながる事が考えられる。これに対して、タンジブル^[1]ユーザインターフェースを取り入れたブロック型プログラミングツール P-CUBE^[2]の開発に取り組んできた。本ツールは、マットにブロックを並べる簡単な操作で、移動ロボットを制御するプログラムを作成できる。ユーザは、P-CUBEを用いて「順次・条件分岐・繰り返し」の3つの基本的なプログラム学習要素を学ぶことができるが、関数の概念を学習することができない。そこで本研究では、P-CUBEの学習要素に「関数」を追加し、その学習効果を検証する。システムには新たに関数ブロック、およびひらがなブロックを導入する。また、音声を出力するロボットを制御対象として新たに導入することで、関数の内容として発話関数

を設定できるようにする。

2. 関数の概念学習に向けたシステムの拡張

プログラミングにおける関数には、キーボードからの入力や画面への出力を行う標準的に準備された関数である `printf()`、`scanf()` などがあげられる。これらの関数を用いるためには、変換指定子や変数の型を覚えなければならない。これに対して本研究では、これらの書式を必要としない関数を取り挙げる。関数ブロックを用いることで「処理の流れの重複を取り除く」、また「一度関数を作成しておけば、後で再利用できる」という利点を学習することを目標としている。今回は、返り値や引数などの関数間のデータのやり取りではなく関数学習の導入として、返り値がないプログラムを設定した。

平成28年に発行された文部科学省が提唱している中等教育一貫の技術分野の教科書^[3]に、プログラムを効率よく記述するためのサブルーチンの使用方法が掲載されている。このように、プログラミング教育に関数の概念が導入されつつある。プログラミングツールにも同様に関数の概念が導入されている。例えば、迷路内のキャラクターの動きを制御できるソビーゴは、キャラクターの一連の動きを関数として設定し、必要な時に呼び出すことができる^[4]。

3. P-CUBE2

3.1 P-CUBE2のシステム構成

関数の学習要素を取り入れた P-CUBE2 を試作した。本

*1: 富山県立大学大学院 工学研究科

*2: 富山県立大学 工学部

*1: Graduate School of Engineering, Toyama Prefectural University

*2: Department of Intelligent Systems Design Engineering, Toyama Prefectural University

ツールでは、新たに関数マット、ひらがなブロック、および関数ブロックを導入し、既存のプログラムマットをメインマットとした。関数マットには、関数ブロックとともにひらがなブロックを並べることで、発話内容を作成できる。図 1 に P-CUBE 2 のシステム構成を示す。プログラミングブロックとプログラムマットのデータ通信には RFID システムを用いることで、無線でのインタフェースを実現し、ユーザはブロックをマットにはめ込むことでプログラム作成操作を完了することができる。RFID(Radio Frequency Identification) とは、電波を用いて RFID タグのデータを非接触で読み書きするシステムのことである。ハンドルには、小型 RFID リーダ ASI4000 を 9 基設置した。また、プログラミングブロックには識別用の RFID タグが内蔵されている。ハンドルをスライドさせることで、関数、およびメインマット上に置かれたブロックの配置情報を読み取ることができる。

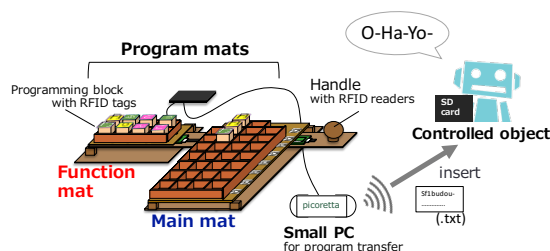


図 1 P-CUBE2 のシステム構成
Fig. 1 System Configuration of P-CUBE2

3.2 P-CUBE2 の操作方法

ユーザには、Main 関数となるメインマットに関数ブロックを用いたプログラムを作成してもらう。関数ブロックの内容は、関数マットにて関数ブロック、およびひらがなブロックを並べることで定義する。

以下に、プログラム作成手順をまとめる。

- ① 関数ブロックを選択する。
- ② 設定したい関数名を関数マットの左端に上面に向けて配置する。
- ③ ひらがなブロックを選択する。
- ④ ひらがなブロックを関数マットに配置した関数ブロックの右側に 3 個置く。
- ⑤ 再度、手順①から④を繰り返し、発話内容を設定する。
- ⑥ 関数マットで配置した関数ブロックと同じ関数名をアルゴリズム構造に基づいて、メインマットに配置する。
- ⑦ プログラム作成後、ハンドルをマットの上部から下部へスライドさせる。
- ⑧ 読み取ったブロックの配置情報を転送用 PC から Wi-Fi を介して、制御対象に転送する。

- ⑨ 制御対象の実行ボタンを押すことで、プログラムを実行する。

3.3 装置構成

プログラムマット

プログラムマットには、関数マット、およびメインマットが用意され、それぞれプログラミングブロックをはめ込む枠が 4 マス×2 段 8、および 30 マス用意されている。関数マット内では、左端に関数ブロックを置き、関数ブロックの右横に 3 個のひらがなブロックを置くことで 3 文字のワードを設定可能である。一つの関数マット上で 2 つの関数を設定することができる。

プログラミングブロック

今回、発話関数を作成するにあたり、ブロックを 4 種類用意した。図 2 に各プログラミングブロックを示す。

ひらがなブロックの形状は、立方体の上面 4 面を面取り加工した形をしている。1 つのひらがなブロックには、ひらがな 50 音のうち、1 行分の文字が割り当てられている。例えば、あ行のブロックは、上面に「あ」、側面は「い」～「お」の 5 文字のひらがなが割り当てられている。各ひらがなブロックの「い」の段にシリコンゴムによる凸情報を付与し、「あ」の段を上面にした状態で時計回りに「う」の段「え」の段「お」の段に当たる文字が存在する。

関数ブロックには F1 から F4 までの関数名が設定されている。形状はひらがなブロックと同様であり、「F1」に凸情報を付与し、「F1」から時計回りに「F2」「F3」「F4」を提示している。

IF ブロックは、制御対象の上部に取り付けられた青いスイッチを ON した場合、および OFF にした場合に、発話内容の 2 つの条件を分岐するために用いる。IF ブロックには、条件分岐の始点を意味する「IF START」と、終点を意味する「IF END」の 2 種類ある。IF START ブロックの上面左側には、EVA スポンジを用いた凸情報を付与することで、スイッチが ON の状態、また、EVA スポンジを貼付しない上面右側はスイッチが OFF の状態に対応することを提示した。

LOOP ブロックは、動作を繰り返す処理をする際に必要なブロックである。他の種類のブロックとの区別をより明確にし、ブロック同士の繋がりを表現するために、LOOP ブロック同士を紐で接続し、一対のブロックとした。

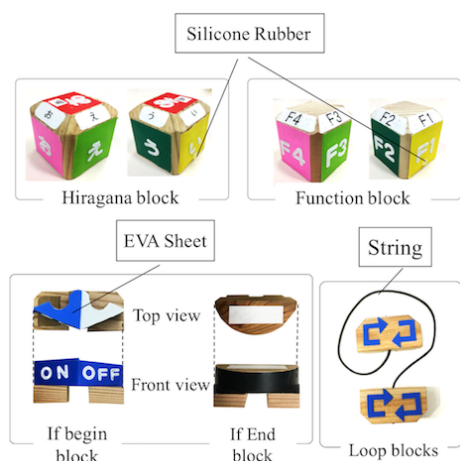


図 2 プログラミングブロック

Fig. 2 Programming Blocks of P-CUBE2.

3.4 学習要素

順次・繰り返し

図 3 に、順次・繰り返しのプログラム例を示す。関数 f_1 に「おはよ」、関数 f_2 に「またね」という発話関数がそれぞれ定義され、関数で設定した発話内容をループさせるプログラムである。ループブロックを用いて、ひらがなブロックあるいは関数ブロックを挟むことで無限ループのプログラムを作成できる。ただし、繰り返し回数の指定はできない。

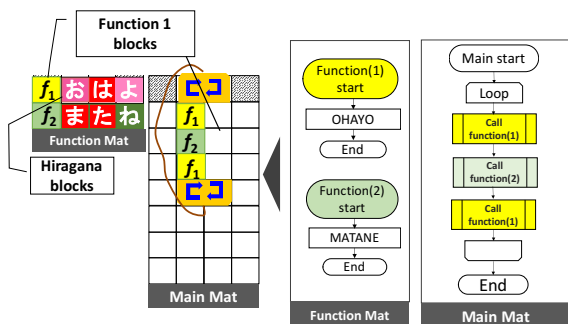


図 3 条件分岐のプログラム例

Fig. 3 Example of a Conditional branch Program

条件分岐

図 4 に条件分岐のプログラム例を示す。関数 f_1 に「おはよ」、関数 f_2 に「またね」という発話関数がそれぞれ定義され、制御対象のスイッチを押すか押さないかで、出力結果を変えるプログラムである。プログラムの出力は、スイッチを押していない間は関数で定義されている「おはよ」を発話し続け、押している間は関数 f_2 で定義されている「またね」を発話し続ける結果となる。

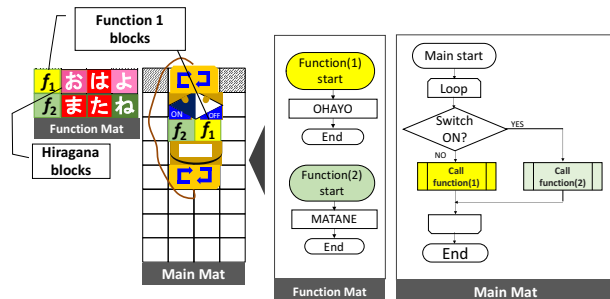


図 4 条件分岐のプログラム例

Fig. 4 Example of a Conditional branch Program

3.5 制御対象

図 5 に制御対象を示す。制御対象に組み込むマイコンボードには Arduino Ethernet を用いた。Arduino には、関数マツで設定した言葉を発話させることができる合成音声 LSI ATP3011F4-PU を組み込んだ。ハンドルに設置された RFID リーダから読み込まれたブロックの配置情報を無線 LAN 経由で転送するため、TOSHIBA 社製の無線 LAN 搭載 SDHC メモリカード Flash Air を使用した。Arduino 上部の青いスイッチは条件分岐ブロックの ON/OFF と対応している。

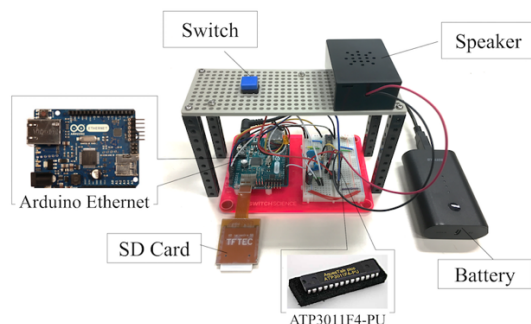


図 5 制御対象

Fig. 5 Controlled object.

4. 使用者に対する印象調査

4.1 課題内容

今回、P-CUBE 2 を使用し、「関数」を用いる利点に気づくことができるか調査した。使用者は、プログラミングの経験がほとんどない 18 歳から 22 歳の男性 6 名、女性 4 名であった。使用者は関数ブロックを使用するグループ（以下、F 群）、および関数ブロックを使用しないグループ（以下、NF 群）に 5 名ずつ分け、図 6 に示す各課題に取り組んだ。課題前、および全ての課題達成後にアンケートを実施した。表 1、および表 2 にそれぞれ事前アンケート、およびツール使用後に実施したアンケート内容を示す。事前アンケートでは被験者の基本的な情報に加え、プログラミングの学習経験やプログラミングに対する苦手意識について調査した。ツール使用後に実

施したアンケートではプログラミングの経験やツールの機構、ツールの使い方、プログラミングに対する興味のほか学習しやすさ、楽しさを調査した。

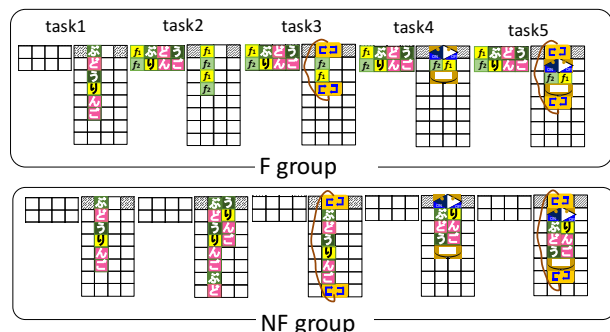


図 6 課題の回答

Fig. 6 Answers of Tasks.

表 1 事前アンケート

Table 1 Questionnaire Before the Experiment

Q1	性別	2件法
Q2	年齢	自由記述法
Q3	プログラミング学習経験	評定法(4段階)
Q4	学習したプログラミング言語	選択回答
Q5	プログラミング経験時「関数」を使用したか	2件法
Q6	関数の使用例	選択回答
Q7	プログラミングに対する苦手意識	2件法

表 2 ツール使用後に実施したアンケート

Table 2 Questionnaire After the Experiment

Q8	ブロックでのプログラミングを楽しめたか	評定法(4段階)
Q9	ツールの操作性	評定法(4段階)
Q10	ツールの操作性の不便さ記述	自由記述法
Q11	プログラミングの構造がイメージできたか	評定法(4段階)
Q12	どの学習要素が理解できなかったか	4件法
Q13	関数の利点を学べたか	2件法
Q14	関数の利点記述	自由記述法
Q15	プログラミングへの興味	評定法(4段階)
Q16	プログラミングに対するイメージ記述	自由記述

以下に、課題内容を示す。なお、NF 群の使用者は、関数ブロックを使用せず、5 つの課題に取り組んでもらう。

課題1. メインマット内にひらがなブロック 3 文字の単語を 2 種類を並べ、発話させるプログラムを作成。

課題2. 関数マット内に関数ブロックと併用してひらがなブロックを並べ、メインマット内で関数を用いることで 3 文字の単語を 2 種を 2 回ずつ発話させるプログラムを作成。

課題3. 発話をループさせるプログラムを作成。

課題4. スイッチが ON, あるいは OFF の時、任意の関数内を発話させるプログラムを作成。

課題5. スイッチを ON, あるいは OFF にしている間、任意の関数内の発話をループさせるプログラムを作成。

4.2 結果・考察

アンケート結果 「関数の利点を学習できたか」という質問に対して、F 群全員が関数の利点を学ぶことができたと回答した。また、関数の利点についての自由記述で得られた回答を以下に示す。

- 略して簡単にできる。
- 数個分のブロックの代用ができる。
- 多くのブロックを使わずにコンパクトに行うことができる。
- F1, F2 を使うと手順が減ってミスが減らせる。
- ブロックの数を減らせる。

これらの結果から、当初の学習目標である「一度関数を作成しておけば、後で再利用できる」という関数を用いる利点を学習することは概ね達成できたと考える。

5. まとめ

P-CUBE に関数マット、および関数ブロックを導入し、P-CUBE 2 を製作した。P-CUBE 2 を用いた印象調査において、関数機能を用いた使用者と関数機能を用いていない使用者を比較した。プログラミングの基本概念である「関数」の利点を学習できたかというアンケートの質問に対して、関数機能を用いた使用者全員が学習できたと回答しており、「一度関数を作成しておけば、後で再利用できる」という関数の利点を使用者が認識していたことが確認できた。

今後、制御対象にセンサを取り付け、2 つの条件分岐ブロックを用いるプログラムなども追加することを検討している。また、ブロックの判別が容易につくようにひらがなブロックの形状を六角柱にする、木材を変えるなど検討したうえで学習効果の検証を継続していく。

参考文献

- [1] 石井裕: タンジブル・ビット- 情報の感触・情報の気配; タンジブルメディアグループ, NTT 出版, (2000).
- [2] 本吉達郎, 掛橋駿, 増田寛之, 小柳健一, 大島徹, 川上浩司: ブロック型プログラミングツール P-CUBE の学習初期段階における有用性の検証; 日本知能情報ファジィ学会誌, Vol. 27, No. 6, pp. 909-920, (2015).
- [3] 新しい技術・家庭 技術分野 未来を創る technology; 東京書籍, p. 241, (2015).
- [4] ソビーゴ子どもブロックプログラミング, <https://hello-sovigo.com/block/>