

# aiBlocks : マイコンにインタプリタを載せて利用する タブレット端末用ビジュアルプログラミング環境

井芹威晴<sup>†1</sup> 光永法明<sup>†2</sup>

近年、タブレット端末が広く普及してきており、パソコンではなくタブレット端末のみを保有する層も増加している。したがって、マイコンのプログラムを作成できるタブレット端末用ビジュアルプログラミング環境があれば、マイコンを使った電子工作を始めやすくなる場合も増えると考えられる。しかし、そのような環境は見当たらない。また、マイコンにインタプリタを載せると、タブレット端末のような資源の限られたコンピュータでも、プログラミング環境を実現しやすい。そこで、iArduino インタプリタの動作するマイコンのプログラムを作成する、タブレット端末用ビジュアルプログラミング環境 aiBlocks を開発したので報告し、製作例としてライトレースカーを紹介する。

## 1. はじめに

近年、タブレット端末が広く普及してきている。タブレット端末はパソコンよりも安価で、軽くて持ち運びやすいものが多い。最近では、パソコンではなくタブレット端末のみを保有する層も出てきており、今後タブレット端末の出荷台数が、パソコンの出荷台数を上回ると予想されている[1]。一方、マイコンが安価になるとともに処理速度や記憶容量といった性能が向上しており、初心者の電子工作でもマイコンを使うことが多くなっている。

マイコンを使った電子工作を初めてする初心者は、プログラミングについても初心者であることが多いと考えられる。プログラミングの初心者は、ビジュアルプログラミング環境を使うとよい [2]。ビジュアルプログラミング環境は、プログラムをテキストの記述ではなく、命令が書かれたブロックを並べるなどして作成する。そのため、キーボードの扱いがおぼつかない子供でも利用しやすく、また、命令の名前や記号の意味を覚える必要がない。そこで、マイコンのプログラムを作成するタブレット端末用ビジュアルプログラミング環境があればよいと考える。

現在、入手しやすいタブレット端末の OS には、Android、iOS、Windows がある。iOS はマイコンのプログラム開発環境実現のハードルが高く、アプリケーションソフトが見当たらない。Windows は、キーボードのあるパソコンだけでなく、タブレット型のパソコンでも利用される OS である。表 1 に Android と Windows 上で動作するマイコン用のプログラム開発環境の例を示す。Android 上で動作するものはテキスト形式の開発環境しか見当たらなかった。Windows 上のソフトウェアはタブレット形式のパソコンでも動作するがマウス操作を中心に考えられているため、必ずしも指で画面を触っての操作はやりやすくない。またタブレット端末での Windows の OS の占有率は 3.7% と低い (2013 年時) [1]。したがって、マイコンのプログラムを開発する、

Android 用のビジュアルプログラミング環境が実現されるとよい。

そこで、初心者にも多く使われている Arduino マイコンボード用の、Android を載せたタブレット端末で動作するビジュアルプログラミング環境 aiBlocks を開発したので報告する。実現には Arduino に iArduino インタプリタを載せ、タブレット端末ではビジュアルプログラミング言語から iArduino インタプリタが解釈・実行できる言語に変換し、コンパイラを利用しない環境とする。以下、第 2 章では aiBlocks の開発の基とした iArduinoTerminal for Android について紹介する。第 3 章では aiBlocks の全体構成と操作方法、実装について説明する。そして、第 4 章で aiBlocks を使った製作例を紹介し、第 5 章で、製作の過程で得た使用感について述べ、最後にまとめと今後の課題を述べる。

## 2. インタプリタ iArduino とターミナルソフト

iArduino は Arduino 上で動作するインタプリタである [5]。Arduino の標準の Arduino 言語に文法等の似たプログラミング言語 iArduino 言語を解釈して実行する。iArduino は非同期シリアルインタフェースを通じて対話的な実行と、実行中のプログラムの制御を受け付け、パソコン上のシリアルターミナルなどを通して操作する。通常のプログラムの実行に加え、ステップ実行、自動ステップ実行 (文の解釈と解釈の間に一定時間実行停止) ができる。そして、iArduino 言語のプログラムを不揮発メモリ (マイコン内蔵の EEPROM) に保存し、リセット時に自動実行ができる。したがって開発が終われば Arduino 単体で動作する。また可読形式に加えバイナリ形式のデバッグ用インタフェースを持っており、入出力ピンの読み書きや変数の値の読み出しがプログラムの実行中でも可能である。加えて実行中のプログラムの位置 (アドレスに相当) を知ることが出来る。

iArduinoTerminal は Windows 用の、iArduinoTerminal for Android は Android 用の iArduino 用ターミナルソフトである。シリアルターミナルに加えて、プログラムエディタ、変数と入出力波形のモニタ、入出力ピンの操作インタフェースを備えている。プログラムエディタはプログラムの実行中の部分をハイライト表示する。これにより、プログラ

<sup>†1</sup> 大阪教育大学大学院  
Graduate School of Education, Osaka Kyoiku University.

<sup>†2</sup> 大阪教育大学  
Osaka Kyoiku University.

表 1 マイコンのプログラムを作成するタブレット端末用プログラミング環境

名称	ハードウェア	動作する OS	記述形式
ArduinoDroid[3]	Arduino[4]	Android	テキスト
iArduinoTerminal for Android[5]	Arduino	Android	テキスト
ArduinoIDE[6]	Arduino	Windows	テキスト
ArduBlock[7]	Arduino	Windows	ビジュアル
Logicator for PICAXE[8]	PICAXE (専用のファームウェアが入った PIC マイコン)	Windows	ビジュアル
LEGO MindStorms[9]	LEGO Group 製のマイコンブロック	Windows	ビジュアル
C-Style[10]	株式会社ダイセン電子工業製のマイコンボード	Windows	ビジュアル

ムの実行の様子, 入出力ピンの様子を難しい操作なしに観察できる. またデバッグ用インタフェース部分のみを Arduino 言語のライブラリとして利用出来るので, iArduino 言語で開発できない速度や規模のプログラムについて Arduino 言語に移行しても, 入出力波形のモニタとピンの操作ができる.

このようにマイコン上のインタプリタと開発用のコンピュータ上でのターミナルソフトを組み合わせることで, 非力なコンピュータであってもマイコン用のプログラムの開発環境を実現できる. そこで aiBlocks は iArduinoTerminal for Android を拡張してビジュアルプログラミング環境を実現する.

### 3. ビジュアルプログラミング環境 aiBlocks

#### 3.1 全体構成

aiBlocks の全体構成を図 1 に示す. プログラムの開発対象である Arduino 上に iArduino インタプリタを載せる. aiBlocks は Android 上で動作し, ブロックを並べて作成したプログラムを iArduino 言語に変換し, iArduino にシリアルインタフェースを通して送る. iArduino は受け取ったプログラムを RAM に記憶し, 逐次解釈しながら実行する. プログラムの開発が終わり RAM 上のプログラムを EEPROM に書き込めば, Arduino だけで動作する. Arduino 上には USB シリアルインタフェースが載っており, バスパワーで動作するのでタブレット端末と USB ケーブルで繋ぐだけでよい.

#### 3.2 画面構成と基本操作

aiBlocks の画面を図 2 に示す. 画面左上にブロックのカテゴリを選択する『カテゴリ選択ボタン』が並び, その下には『ブロックパレット』がある. 画面中央には, ブロックを並べてプログラムを作る『プログラムエリア』があり, その上部には実行に関するボタンが並び『実行ツールバー』がある.

プログラムを作成するときはまず, 使いたいブロックのカテゴリをタップする. すると, 選択したカテゴリのブロックパレットが表示される. その中から使いたいブ

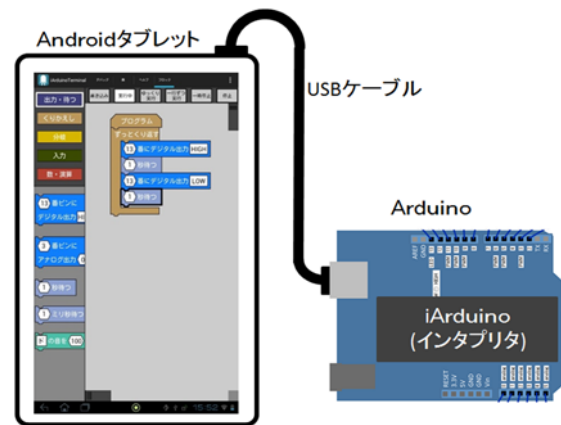


図 1 aiBlocks の全体構成

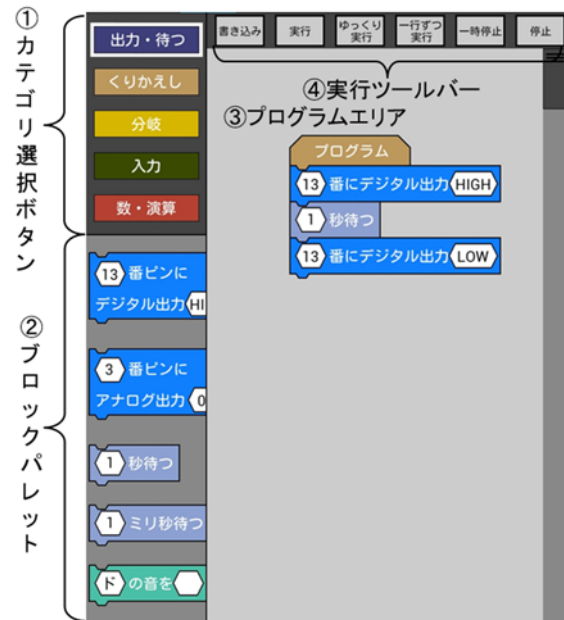


図 2 aiBlocks の画面構成

ロックを選び, プログラムエリアにドラッグし並べる. 図 2 では「出力・待つ」のカテゴリの「デジタル出力」ブロックと「待つ」ブロックを並べて, 1 秒間 Arduino 上の LED を点灯するプログラムを作っている. プログラムが完成したら, 実行ツールバーの『書き込み』ボタンをタップする. するとブロックのプログラムが iArduino 言語に変

換され、iArduino に送られる。『実行』ボタンをタップすると、そのプログラムが実行される。

### 3.3 aiBlocks のプログラムブロック

aiBlocks で用意しているブロックをカテゴリ毎に、表 2 に示す。ブロックのカテゴリは、『出力・待つ』『くりかえし』『分岐』『入力』『数・演算』の 5 種類である。ブロックは全部で 19 種類である。『出力・待つ』には、デジタル/アナログ出力と待つ (busy loop)、音を出すブロックがある。『くりかえし』には、C 言語の for 文と while 文に相当するブロックが、『分岐』には、if 文に相当するブロックと、条件式でよく利用する演算子に相当するブロックがある。『入力』には、デジタル/アナログ入力のブロックが、『数・演算』には、数値や変数、代入、四則演算のブロックがある。

ブロックの形状は、単独で C 言語の文 (式文) に相当するブロック (デジタル出力や変数への代入など)、式の一部になるブロック (数値、変数名、演算子、デジタル/アナログ入力など)、制御文に相当するブロック (for, while, if 文など) で変え、形状が異なるブロックが結合できないようにしている。これにより文法的な誤りを防ぐ。

また、数値や変数名などの入力には専用の入力パッドを用意している。数値・変数名の入力パッドを図 3 に示す。上部のタブで数値と変数名の入力モードを切り替える。(a)数値モードでは 8 桁までの数値を、(b)変数名モードでは、a から m の範囲のアルファベット 1 文字を入力できる。他

に、ピン番号、アナログ入力の値、等号・不等号、音階、四則演算記号の入力用パッドがある。これにより、正しい iArduino 言語として変換できない、あるいは実行してもおかしくなるような文字入力を出来るだけ防ぐ。

aiBlocks で作ったプログラムの例を図 4 に示す。『無限ループ』のブロックの中に、『デジタル出力』と『待つ (秒)』のブロックがある。このプログラムを iArduino 言語のプログラムに変換すると、図 5 になる。このプログラムはユーザには見せないが、『書き込み』ボタンをタップするとこの変換をして iArduino に送る。これを実行すると Arduino 上の LED (13 番ピンに接続) が 1 秒ごとに点灯/消灯を繰り返す。



(a)数値モード (b)変数名モード  
 図 3 数値・変数名の入力パッド

表 2 aiBlocks のブロック

カテゴリ	名称/C 言語の相当名称	動作・処理等
出力・待つ	デジタル出力	指定したピンにデジタル出力する
	アナログ出力	指定したピンにアナログ出力 (256 段階の PWM 出力) する
	待つ (秒)	指定した時間 (秒単位) だけ待つ
	待つ (ミリ秒)	指定した時間 (ミリ秒単位) だけ待つ
	音	指定した周波数 (Hz) の音を指定した時間 (ミリ秒単位) 鳴らす
くりかえし	無限ループ	ずっとブロック群を繰り返す
	for 文	指定した回数ブロック群を繰り返す
	while 文	条件式が成り立つ (真: 値が 0 でない) 間ブロック群を繰り返す
分岐	if 文	条件式が成り立つ (真: 値が 0 でない) ときブロック群を実行する
	if - else 文	条件式が成り立つとき 1 つ目のブロック群を、成り立たないとき 2 つ目のブロック群を実行する。
	比較演算子	左右の値を比較し、等式 (不等式) が成り立つとき 1 となる演算子
	AND	左右がともに真である場合 1 となる演算子
	OR	左右のいずれかが真である場合 1 となる演算子
入力	デジタル入力	指定したピンのデジタル入力の値 (0 か 1) を返す
	アナログ入力	指定したピンのアナログ入力の値 (0 ~ 1023) を返す
数・演算	定数・変数	定数 (数値), あるいは変数名 (アルファベット a ~ m) を表す
	代入文	変数に式の値を代入する
	四則演算	四則演算 (二項演算子)

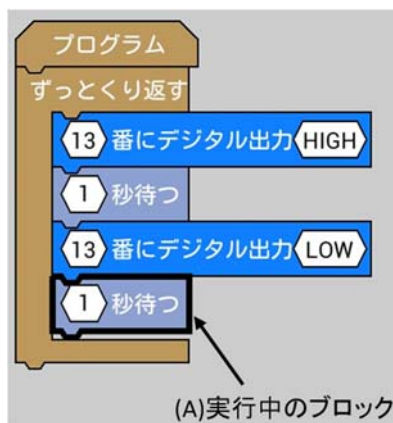


図 4 aiBlocks で作ったプログラムの例

```
pinMode(13,OUTPUT);
for(;;){
    digitalWrite(13,HIGH);
    delay(1*1000);
    digitalWrite(13,LOW);
    delay(1*1000);
}
```

図 5 図 4 を変換した iArduino 言語のプログラム

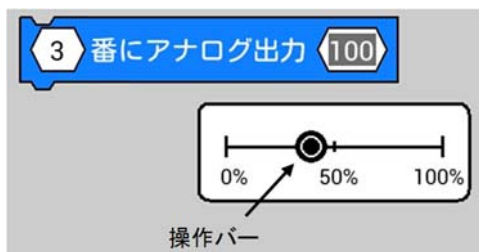


図 6 アナログ出力のブロックとその入力パッド。操作バーをスライドさせると、出力が変化する。

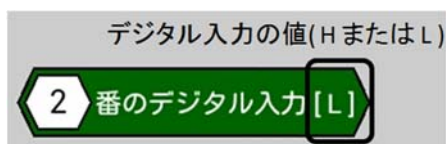


図 7 デジタル入力のブロック

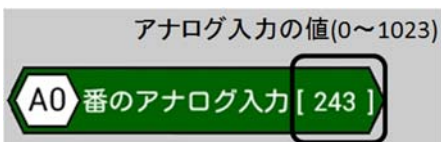


図 8 アナログ入力のブロック

また出力に関するブロックをタップすると、すぐにピンの出力が変わる。例えば、図 4 のデジタル出力のブロックをタップすると、13 番ピンの出力がすぐに HIGH または LOW になる。また、アナログ出力の値の入力パッドは、操作バーをスライドさせて値 (0~255) を変える (図 6)。その際、指をバーにのせたままでも出力が変化する。

入力のブロックは、入力の値をリアルタイムに表示する。デジタル入力のブロックは、電圧が高い場合は H、低い場合は L をブロックに表示する (図 7)。Arduino のアナログ入力は、0~5V の電圧を 1024 段階で測ることができる。アナログ入力のブロックは、測った電圧を 0~1023 の数値で表示する (図 8)。

### 3.4 プログラムの実行

プログラムの実行中は、実行しているブロックをハイライト表示する。図 4 では、(A)の 1 秒待つブロックを実行中である。プログラムの動作を観察しやすいよう、通常の実行 (図 2 の「実行」ボタン) に加え、一命令実行すると停止するステップ実行 (「一歩ずつ実行」ボタン) と、文の解釈と解釈の間に 0.5 秒間実行を停止する自動ステップ実行 (「ゆっくり実行」ボタン) ができる。

### 3.5 ブロックの描画と操作の実現

aiBlocks の開発言語は Android 標準の Java である。ブロックの描画に適した標準のライブラリが無いため、画面の図 2 の部分を Canvas とし、そこに多角形や円弧の描画によりボタンやブロックを実現している。そのため、プログラムエリアのスクロールも aiBlocks アプリ内に実装して実現している。Canvas 内をタップすると、タップした座標によりブロック上、プログラムエリア内 (ブロック上ではない)、ボタン上、それ以外 (何もしない) と判断する。ブロック上をタップしたと判断した後に、指を動かすとドラッグと判断し、指の動きに合わせてブロックの座標を更新し、Canvas を再描画する。指が離れたら (ドラッグ終了)、離れた場所の座標と、すでにプログラムエリアにある各ブロックの座標を比較し、隣接ブロックを探す。ドラッグしたブロックが既存のブロックの下より (文に相当するブロックの場合) で、座標のずれがブロックの大きさの 1/3 以内であれば隣接ブロックとし、既存のブロックとつなぐ。

ブロック上ではないプログラムエリア内をタップした場合には、スクロールをしたいと判断し、ブロック全体の描画原点を移動してスクロールの効果を出す。それ以外のカテゴリ選択ボタンや実行ツールバー上のボタンのタップであれば、それぞれの機能を実現する関数を呼び出す。

実行中のブロックのハイライト表示については次のようにして実現している。『書き込み』ボタンがタップされ、ブロックによるプログラムを iArduino 言語に変換する際に、各ブロックの変換後のプログラムリスト上での位置 (先頭からのバイト数と長さ) を記憶しておく。プログラムの実行中には iArduino から実行中のプログラムの位置が送られてくるので、対応するブロックを探し、それをハイライト表示 (枠線を太く) する。

## 4. aiBlocks を使った製作例

aiBlocks を使って製作したライントレースカーの外観を図 9 に、回路図を図 10 に示す。マイコンボードは

ArduinoUno, 電源には単三形乾電池 4 本を使用する。モータの駆動にはトランジスタを使用し, ベース端子をアナログ出力ができる 10 番と 11 番ピンに接続する。ラインの判別には LED と CdS セルを使用する。LED と CdS を, ライン(黒色)の幅以上のスペースを空けて 2 組固定し, LED で床面を照らし, CdS でその反射光を受ける。固定抵抗と CdS の抵抗分圧により, CdS の明るさの変化による抵抗値の変化を電圧に変える。CdS で受ける光が弱い(ライン上)とアナログ入力の値が大きくなり, 逆に受ける光が強い(ライン外)とアナログ入力の値が小さくなる。

車体の中心がライン上にあるときは左右の CdS の間にラインがあるので, アナログ入力の値は左右ともに小さい。このときは左右のモータの回転数を同一にして直進すればよい。ラインが左に曲がっていたり, 車体の中心がラインの右寄りになり, 左の CdS がラインの黒色の影響を受けると, アナログ入力の値が大きくなる。このとき左のモータの回転数を下げれば, 車体の中心がライン上に戻る。右の CdS がラインの黒色の影響を受けた場合も同様である。

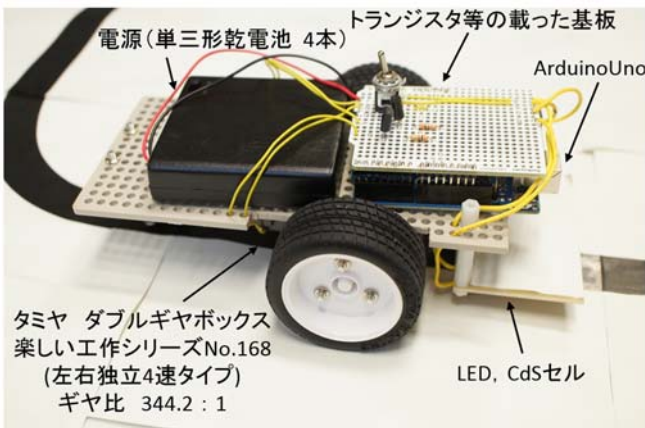


図 9 ライントレースカーの外観

ライントレースのプログラムを図 11 に示す。車体の左にある CdS を A0 番ピンに, モータを 10 番ピンにつないでいる。一つ目の if - else 文で, CdS が黒色の影響を受け A0 番ピンの値が 250 以上であれば, 左のモータの回転速度を決める 10 番ピンのアナログ出力を 15 とし, そうでなければ 50 としている。二つ目の if - else 文は, 車体の右にある CdS とモータ用である。この二つの if - else 文を繰り返すことで, ライン上を左右に振れながら車体が進む。

### 5. 使用感

ライントレースカーは筆者の一人が製作したが, その過程で以下の使用感を得た。こういった小さな模型を初心者が製作する場合には, if - else 文の条件式の閾値を, ライン



図 11 aiBlocks でのライントレースのプログラム

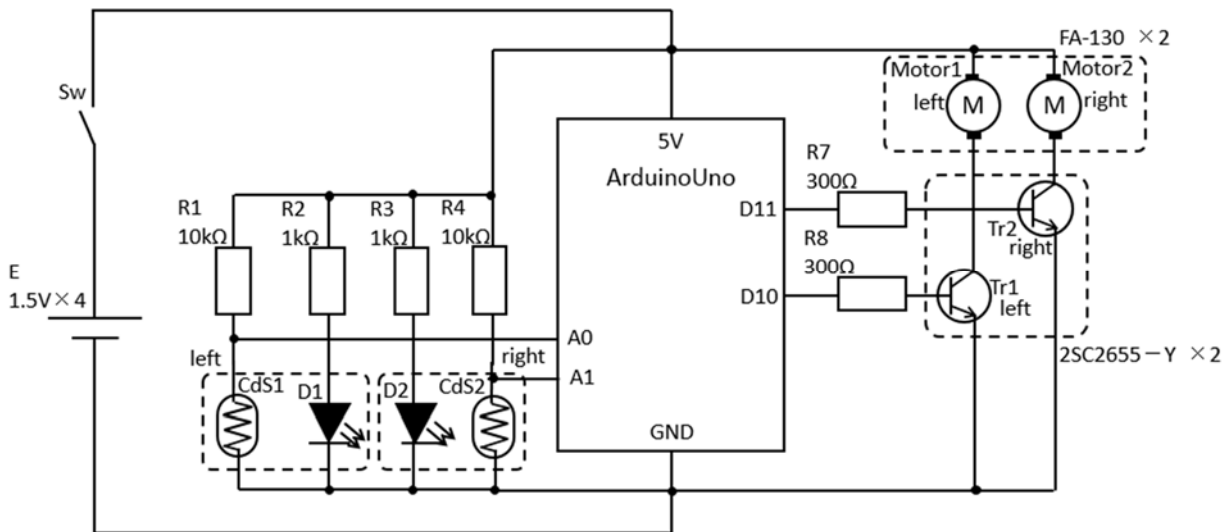


図 10 ライントレースカーの回路図

とライン以外の反射率などから計算によって決めるのではなく、実際にラインとライン以外でのアナログ入力の値を測って決めることが想定される。そのような方法で閾値を決めるにあたり、アナログ入力の値をブロックに表示していることが有効と感じた。また、モータの回転数も試行錯誤的に決めると想定される。それについても、アナログ出力のブロックをタップして二つの速度の切り替え、入力パッドの操作バー（図 6）による実際のモータの回転数の調整が有効と感じた。

一方で、ブロックの移動操作が直感的でないと感じる場面があった。具体的には、if 文などのブロックを移動させようとタップしてドラッグしたつもりが、条件式のブロックをタップしてドラッグしてしまうことが何度もあった。図 12 の if - else 文のブロックを移動させようとしたときには、条件式の部分ではなく、「もし」「なら」といった文字の部分の部分をタップしてドラッグする必要がある。しかし、条件式を含んだブロック全体の中央、つまり条件式の中央をタップする傾向が強かった。同じ筆者が Scratch や ArduBlock をマウスで操作した場合には、そのような傾向がなかったことから、指によるタッチパネル操作固有の問題の可能性がある。

## 6. まとめ

本論文では、インタプリタを載せたマイコンのプログラムを作成する、タブレット端末用ビジュアルプログラミング環境 aiBlocks について報告し、製作例としてライトレースカーを紹介した。今後は、サンプルプログラムを充実させるとともに、操作性の問題を解決したい。本研究は JSPS 科研費 25870418 の助成を受けたものである。



図 12 条件式を含んだ if - else 文ブロック

## 参考文献

- 1) 総務省：“平成 25 年版情報通信白書 第 1 部 特集「スマート ICT」の戦略的活用でいかに日本に元気と成長をもたらすか”詳細 (<http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h25/html/nc112112.html>)
- 2) 兼宗，阿部，原田：“プログラミングが好きな言語環境”，情報処理，Vol.50，No.10，pp986-995(2009).
- 3) Google play：“ArduinoDroid-ArduinoIDE”，詳細 (<https://play.google.com/store/apps/details?id=name.antonsmirnov.android.arduinoandroid2&hl=ja>)
- 4) Arduino project：“Arduino”，詳細(<http://www.arduino.cc/>)
- 5) 光永法明：“タブレット端末で動作する，インタプリタ型言語搭載マイコンのプログラミング環境の開発”，情報処理学会研究報告，Vol.2013-CE-119，pp1-4(2013).
- 6) Arduino project：“Arduino IDE”，詳細 (<http://arduino.cc/en/Main/Software>)
- 7) ArduBlock：“ArduBlock”，詳細(<http://blog.ardublock.com/>)
- 8) Revolution Education Ltd：“Logicator for PICAXE”，詳細 (<http://www.picaxe.com/Software/PICAXE/Logicator-for-PICAXE/>)
- 9) LEGO Group：“LEGOMINDSTORMS”，詳細 (<http://www.lego.com/ja-jp/mindstorms>)
- 10) 株式会社ダイセン電子工業：“TJ3-B 用 C-Style”，詳細 ([http://www.daisendenshi.com/view\\_download.php?master\\_category\\_s\\_id=145&products\\_id=447&products\\_name=%8E%A9%97%A5%8C%5E%83%8D%83%7B%83b%83g%90%BB%8D%EC%83L%83b%83g%20TJ3B](http://www.daisendenshi.com/view_download.php?master_category_s_id=145&products_id=447&products_name=%8E%A9%97%A5%8C%5E%83%8D%83%7B%83b%83g%90%BB%8D%EC%83L%83b%83g%20TJ3B))