

多様な学習形態を可能とするLED照明向け プログラミング教材の開発

大見 嘉弘* 永井 保夫*

LEDを用いた照明を制御できるプログラミング教材を開発した。本教材はソフトウェア開発者を育成する大学の課程においてプログラミング能力を育成することを目的とする。対象とする照明機器は10本の柱状照明から構成されており、カードサイズの小型コンピュータからそれぞれの照明の発光色を制御できる。また、各照明に距離センサを内蔵しており、照明の前に人が立つなどすると反応する作品が開発可能である。本教材ではPC上で動作するプログラムの開発環境、LED照明のエミュレータ、小型コンピュータ上で動作するプログラムの開発環境、および本格的なLED照明と簡易装置を組み合わせることで、5種類の学習形態を提供した。これにより、初学者向けの開発環境から始め、徐々に高度な開発環境を利用することが可能となり、円滑なプログラミングの学習が期待できる。高度な開発環境を利用することで、組み込みシステムの入門学習を行なうことができると考える。

キーワード：プログラミング教育、LED照明、組み込みシステム、適応処理、フィジカルコンピューティング

A Development of Teaching Material of Programming for LED Lighting Which Supports Various Learning Styles

Yoshihiro OHMI* and Yasuo NAGAI*

A teaching material for programming which is able to control LED lighting is developed. The purpose of this material is training of programming skills in the course of software development. This material consists of ten pillar type lights, and each light color can be controlled from a card-size microcomputer. Each light module has a distance sensor, and students can develop some works which response by human behavior around these lights. This material has five typed learning styles, such as runnable program on PC, using an emulator of LED lighting and runnable program on the card-size microcomputer. By starting to use the development style for beginners and stepping up to the advanced style, it is expected that students can learn smoothly. When students use the advanced style, it is expected that this material introduce development of embedded systems.

Keywords: education of programming, LED lighting, embedded system, adaptive processing, physical computing

1. はじめに

近年、旧来のキーボードやマウスを用いたキャラクターユーザインタフェース (CUI) やグラフィカルユーザインタフェース (GUI) だけでなく、コンピュータと物理的な世界を積極的につないだインタフェースが身近になりつつある。例えば、加速度センサやカメラで人の動きを検出して操作する方法や、カメラやGPS等で周辺の実世界の情報を捉え、その情報にPCで生成された情報を重畳する拡張現実感などが挙げられる。また、スマートフォン等の普及により、タッチパネルが身近なものになってきている。

これに対して、大学等における従来のプログラミング教育では、演算やファイル処理などの情報処理に重点を置き、プログラムが扱う入出力に関する処理には比重が置かれていない場合が多い。このため、もっぱらCUIやGUIのみを取り扱い、近年、身近になった新たなインタフェースは扱わない場合が多く見られる。このような、CUIやGUIのみのプログラミング教育では、学生の興味と教育内容の間に乖離が進み、学習意欲の低下や挫折を招くおそれがあると考えられる。今後、CUIやGUIに限定されない新たなインタフェースを用いたシステム開発が増加すると予想され、新しいインタフェースを用いるプログラミングを積極的に教育に取り入れる必要がある。

そこで、このような新しいインタフェースのひとつとして、マイクロプロセッサボードもしくはPCで制御できる本格的なLED照明機器を用意して、制御プログラムを開発するための教材を開発した。教材の対象としてLED照明を採用した理由を以下に列挙する。

- 発光色を高速に様々に変更できるため、複雑な視覚的表現が可能であり、観賞用の照明として深い印象を与えられる。
- センサを用いることで、近傍の人に反応するインタラクティブな照明が作成できる。

- 照明をどのように光らせるかは、プログラミング次第であり、学生が意欲を持ってプログラミングを学ぶことができる。
- 照明はあらゆる人に身近なものでありながら、照明を用いたプログラミング教育はほとんど行なわれていない。このため、身近な照明を用いたプログラミングは多くの学生が興味を持つと考える。

本論文では文献[1]の成果を利用して、より多くの初心者向けのプログラム開発方式を提供することで、プログラミングが苦手な学生でも取り組みやすく工夫した成果を示した。

本論文で提案した教材は、以下の4つから構成される (図1)。以下の各項目の番号は図1中の番号に対応する。

(1) LED照明機器

センサを内蔵しており、人間と照明機器間との距離を検出できる。また、24ビットカラーの色表現が可能で、鑑賞用の照明としては十分な光量がある。

(2) 簡易LED照明機器

(1)の簡易版として、卓上でプログラム開発が行える機器である。

(3) プログラム開発環境

組み込みシステムとしては親しみやすい開発環境を用いている。また、文献[1]で示した成果に加え、PCからLEDの発光色を直接制御したり、画面上に仮想的な照明を表示するエミュレーション機能を設けることで、組み込みシステム開発より容易にプログラム開発が行なえる方法を用意した。

(4) サンプルプログラム

開発が容易になるよう、典型的なプログラムや、複雑な表現を行うプログラムを例として用意した。

以上で示した教材の多くは、文献[1]で示した成果をそのまま使用している。本論文で示す新たに開発した項目は、(3)のプログラム開発環境における「PCからLEDを直接制御する機能とLED照明のエミュレーション機能」である。

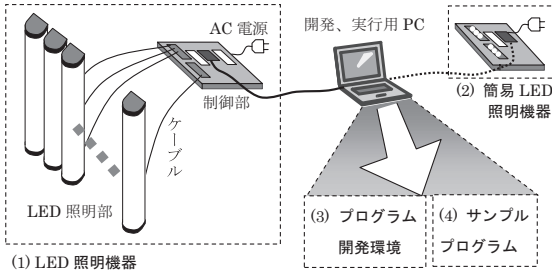


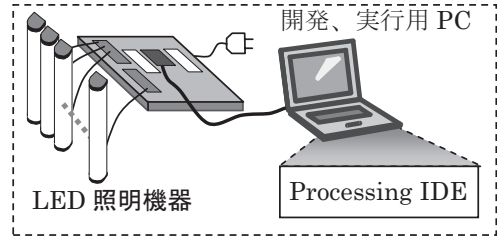
図1 LED照明教材の構成

図2に、本教材が提供する5種類の開発方式を示す。なお、図2の(A)、(A')、(B)は本論文で示す新たに開発した方式であり、(C)、(C')は文献[1]で示した成果である。本教材を用いると最初は、開発が容易なPC上で動作するプログラムの開発から始めることができ(図2の(A)、(A')、(B))、最終的にはマイクロプロセッサボードで動作するプログラムの開発(図2の(C)、(C'))を習得することが可能となるため、組み込みシステムにおけるソフトウェア開発の学習が円滑に行える。また、組み込みシステムの学習が必要ない場合は、PC上で動作するプログラムの開発のみを行ない、プログラミングの能力を高めていくことも可能である。今後、この教材を用いて、センサの入力に応じた照明が変化する芸術的な作品を学生が作成することで、学生のモチベーションを高め、創造的な姿勢の醸成やプログラミング能力の向上を図ることが期待できる。

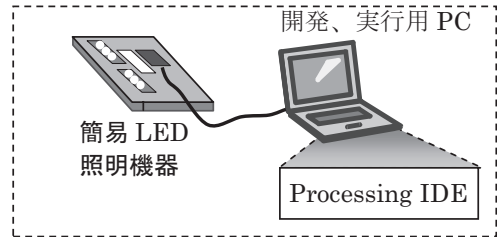
2. LED照明機器の設計と実装

2.1 LED照明機器の概要

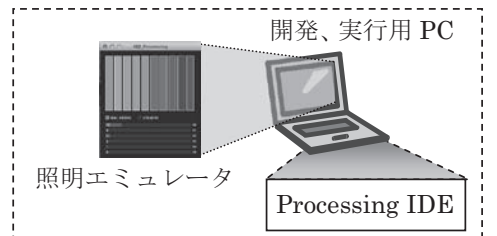
開発したLED照明機器の構成図を図1内の(1)に、実際の照明部を図3に示す。照明は10本の柱状のLED照明であり、それぞれ独立して床の自由な場所に置くことを想定している。それぞれのLED照明は、高さが1m程度であり、RGBごとに12bit、計36bitカラーで色を指定することができる。ただし、後述するように本教材では一般的な24bitカラーで指定するようにしている。それぞれのLED照明はケーブル



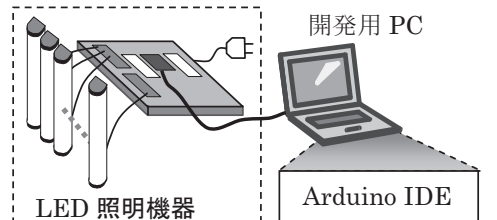
(A) LED照明のPC制御プログラミング (プログラムはPC上で動作する)



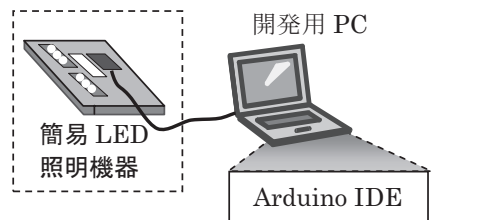
(A') LED照明のPC制御プログラミング (プログラムはPC上で動作する)



(B) エミュレータを使用したPC制御プログラミング (プログラムはPC上で動作)



(C) LED照明の組み込みプログラミング (プログラムはPCなしで動作する)



(C') LED照明の組み込みプログラミング (プログラムはPCなしで動作する)

図2 本教材の5種類の開発方式

ルで制御部に接続される。制御部には、マイクロプロセッサが含まれており、PCで作成したプログラムをUSBで転送して動作させることが可能である。さらに、それぞれのLED照明には、距離センサが内蔵されており、制御部に結線することで、人間の動きなどに応じて照明を変化させるような一種のインタラクティブアートを実現することができる。また、これらのLEDの発光色の制御と距離センサの値の取得は、PCからリアルタイムに行なうことも可能である（3.1節参照）。

以下で、LED照明機器の構成要素について詳細に説明する。ただし、文献[1]で述べた事項の多くは省略する。

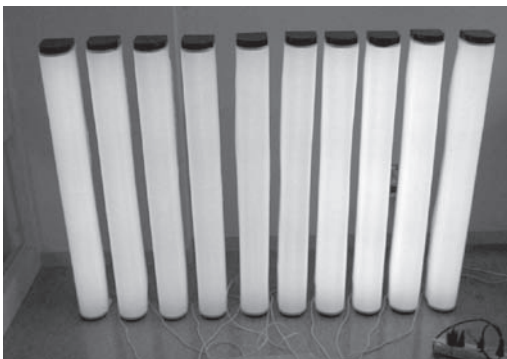


図3 LED照明機器の照明部の写真

（実際は1本ずつ違う色で、全体として虹色に表示されている）

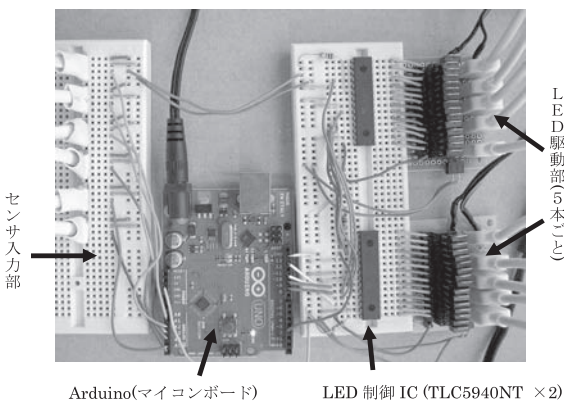


図4 制御部の写真

2.2 制御部の設計と実装

照明部10本への電源供給、発光色の制御、センサからの入力を受け入れる制御部について説明する。開発した制御部の写真を図4に示す。

ソフトウェアによってセンサ入力を受け入れ、照明部の発光を制御するためにArduinoを使用した[2]。Arduinoは小型のマイクロプロセッサボードであり、従来の同種のボードと比較してソフトウェア開発が容易に行なえるのが特徴である。このArduinoを用いた本教材のソフトウェア開発方法については、3節で述べる。表1に本教材で使用したArduinoの一機種であるArduino Unoのハードウェア仕様を示す[3]。

照明部には距離センサが1個ずつ付いているため、計10個のセンサが備わっているが、Arduino Unoはアナログ入力が6本であるため、最大6個のセンサを接続して使用できる。6個より多くのセンサを使用する場合は、アナログ入力が16本あるArduino MEGA2560[4]を使用する。

2.3 簡易LED照明機器の開発

本照明機器の照明部は、10本全て並べると高さ1 m、幅1 m以上の場所をとる。さらに、照明部は光量が大きいため、プログラム開発中の動作確認で用いると、予期しない動作で激しいちらつきを生じ、てんかんに類する症状を発する恐れがある。このため、本照明機器とは別に、照明機器の簡易版を開発した（図1、2、5）。

表1 Arduino Unoの主なハードウェア仕様

プロセッサ	Atmel ATmega328P
クロック	16MHz
動作電圧	5 V
デジタルI/O	14本
PWM出力	6本
アナログ入力	6本
プログラミング可能容量	31.5kB
SRAMメモリ容量	2 kB
EEPROM容量	1 kB

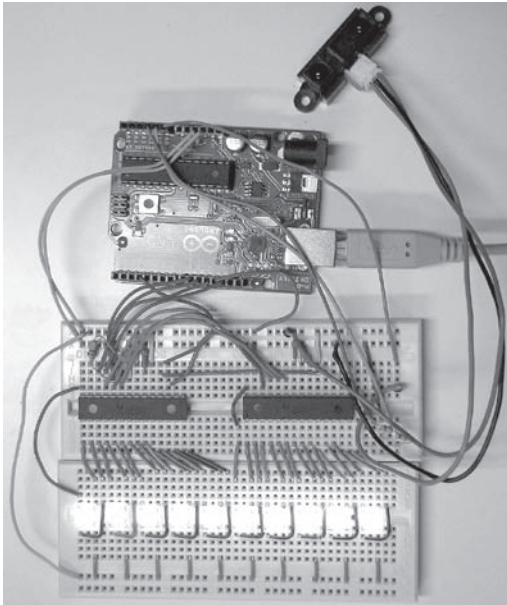


図5 簡易LED照明機器の写真

これは、ブレッドボード上にRGB LEDを10個並べたもので、LED 1個が照明部1本に相当する。光量が小さいため、激しくちらつく場合でも視覚的な悪影響は小さい。また、部品点数が少ないため、製作も容易で、USBバスパワーで動作するため手軽に使用できる。学生がプログラムを開発する際は、原則としてこの簡易版を使用して動作確認を行い、完成に近づいた時点で、本照明機器に移行することで、複数学生による開発が円滑に行え、光量による悪影響を抑えることができる。なお、文献[1]の後に配線の再検討を行い、使用するブレッドボードを3枚から2枚に減らしたため、より小さく扱いやすくなった。

3. プログラム開発環境

本教材のプログラム開発ではPC上で動作するプログラムの開発(図2の(A)、(A')、(B)に相当)にProcessingを、組み込みプログラムの開発(図2の(C)、(C')に相当)にArduino IDEを用いる。以下で、それぞれについて説明する。

3.1 PC上で動作するプログラムの開発環境

初学者向けに、PC上で動作するプログラムからArduinoを制御する手段を開発した(図2(A)と(A'))。PC上で動作するプログラムの開発を行うためにProcessingを用いる。図6にProcessingの画面例を示す。Processingは画像、アニメーション、インタラクションを扱うプログラムを手軽に作成できるプログラミング言語・環境である[5][6]。基本的にはJava言語であるが、クラス定義やmainメソッドの定義を省略でき、グラフィックス表示などの、頻繁に利用する機能を簡便で分かりやすく書くことができるライブラリが提供されている。内部的には「Javaへの変換→Javaのコンパイル→実行」という過程を経てプログラムが実行されるが、シンプルな外観のツールで、Runボタンを押すだけで上記の過程が実行される。そのためスクリプト言語で書くのに近い感覚でプログラミング作業が行える。そして、フリーソフトウェアであるため、教育機関で導入しやすい。東京情報大学情報システム学科では、3年次にProcessingを一貫して用いる授業が用意されているため、本教材によるLED照明のプログラ



図6 Processingの画面例

ミングが円滑に学べる。

PCからArduinoを制御するには、何らかのプロトコルに基づいた送受信プログラムをPCとArduinoの双方に用意する必要がある。通常、PCとArduinoの間の接続はUSBを用いる。このためUSB上の仮想シリアル通信による送受信プログラムを作成することが常套手段である。しかし、本研究では独自のプロトコルを用意せず、Firmata[7]を拡張して使用することにした。その理由は、FirmataがArduinoをPCから制御するために広く用いられているプロトコルであり、本研究で独自の仕組みを新たに設けるよりも、Firmataの既存の機能を流用し、LED照明の制御機能を追加したほうが有益であると判断したためである。標準のFirmataではデジタル入力、アナログ入力ポートの値の読み込み、デジタル出力、アナログ出力ポートへの出力値設定がPCからリアルタイムで行なえる。

このFirmataに、本照明で使用するLED駆動ICを扱う機能を追加するために、標準のFirmataを拡張した。拡張には標準のFirmataに備わっている機能を損なわないように、拡張用のSYSEXコマンドの一種のExtended Analogメッセージを利用した。図7にLED照明の発光色を指定するプロトコル例を示す。このメッセージ1つで、Light Channelで指定した1個のLED照明の発光色（24bitカラー）を変更する。Extended Analogメッセージはデータ部が可変長であるが、現在のFirmataの実装ではデータ部が3バイトに固定されている。このた

Byte

1	START_SYSEX (0xF0)
2	EXTENDED_ANALOG (0x6F)
3	Light Channel Number (0...127)
4	Red Color value (8...2 bit)
5	Green Color value (8...2 bit)
6	Blue Color value (8...2 bit)
7	RGB LSB value (1bit, 1bit, 1bit)
8	END_SYSEX (0xF7)

図7 LED照明制御用プロトコル

め本照明の制御ではデータ部を5バイトとし、Arduino側のFirmataプログラムでデータ部のバイト数に応じて標準の制御と本照明の制御に分けるように実装した。このため、本来の実装されているメッセージはそのまま使用できる。なお、2.1節で述べたように本LED照明は36bitカラーの発色が可能であるが、これは一般的ではないため、24bitカラーで指定するようにした。

なお、LED照明に内蔵する距離センサの値の取得は標準のFirmataに備わっているアナログ入力機能を利用した。その他のFirmataの機能については、LED照明機器の制御に用いている3、9～13のデジタルポートの入出力機能を除いてそのまま利用可能である。

さらに、LED照明のハードウェアがない状況でもプログラム開発が行えるように、エミュレータを開発した（図2(B)、図8）。次節で示すプログラム教材を使用すると、プログラム起動時にこのエミュレータ画面がPC画面に表示される。エミュレータ画面の上部には10本の照明部の発光色が表示される。また、下部のバーは照明部の距離センサに対応しており、EMULATIONモードの場合は、マウスで

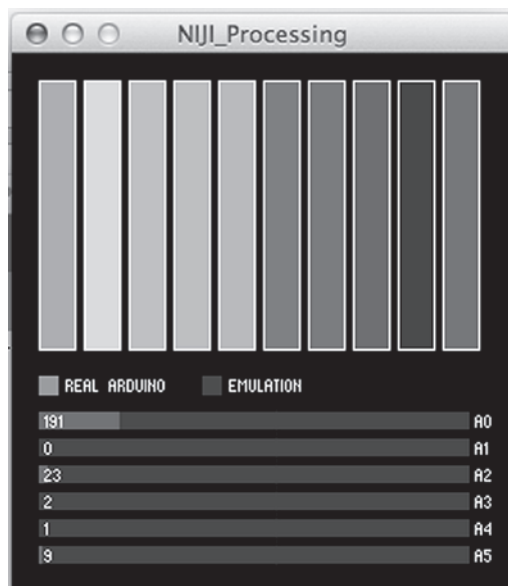


図8 LED照明エミュレータの画面

スライドすることで距離センサの値を自由に調整できる。これは、PCにLED照明機器が接続されていない場合に、実際の距離センサの代わりに使用する。さらに、下部のバーは距離センサのモニタ機能を兼ねている。PCにLED照明機器が接続されている場合は、エミュレータをREAL ARDUINOモードにすると、実際の距離センサの値が下部のバーにリアルタイムで反映される。このため、実際のセンサ値を確認しながらプログラム中のしきい値等を決める作業が楽に行なえる。

3.2 組み込みプログラムの開発環境

PCなしで、Arduino単独で動作するプログラムの開発(図2の(C)もしくは(C'))に相当)にはArudino IDEを用いる。この開発環境は文献[1]に示した成果である。Arduino IDEはプログラミング開発環境Processingを元に開発されている[2][6]。Arduino IDEはProcessingに極力似せて設計されているため、従来の組み込みシステム開発と比べ格段に容易に開発が行えるという特長がある。ProcessingはJavaをベースとし、Arduino IDEで用いるArduino言語はC言語をベースとしているという違いがあるが、その違いも極力吸収するような工夫がなされている。例えば、プログラムの実行モデルが共通である。双方の初期化のメソッド(関数)は`setup()`であり、その後Processingでは`draw()`が、Arduino言語では`loop()`が定期的に呼び出されることで処理が進む。また、Arduino言語を用いる大多数の用途ではC言語のポインタを扱う必要がないため、Javaの文法であるProcessingとほぼ同じ記述となる。本教材でセンサ入力や照明の制御を扱うプログラムにおいても、C言語に特有なポインタを扱うコードを記述する必要がない。一方、組み込み向けプロセッサを用いる場合、C言語においても一部のコードをアセンブラで記述したり、レジスタの設定を行う必要があることが少なくない。これに対して、Arduino言語ではそのような組み込み向けプロセッサ特有の仕様が巧妙に隠されて

おり、PC上で動作するプログラムの開発とほぼ変わらない感覚でコーディングが行える[2][8]。

このように、Arduino IDEを用いると、従来の組み込みシステム開発と比べ格段に開発が容易であるが、Processingを用いた開発に比べると困難な点がある。一番の困難な点はデバッグが難しいことである。Arduinoでは、デバッグが用意されていないため、プログラムの動作を確認するために、プログラム中の要所に変数の値を表示する命令を入れるのが一般的である。Processingであれば、`println`メソッドを使用するだけで良いが、Arduinoでは、接続したLEDや液晶モジュールに表示するか、仮想シリアル通信でPCに出力する必要がある、手間がかかる。また、ProcessingはJavaがベースであるため、プログラムに問題がある場合、例外が発生してバグがある箇所を特定しやすい傾向にあるが、Arduino言語はC言語がベースであるため、特に配列やポインタを使った場合、バグの特定が非常に困難になることがある。これらの原因は、組み込みシステム特有の事情とベース言語の違いによるものである。

このように、Arduino上で動作するプログラムの開発はPC上で動作するプログラムの開発にない困難さがあり、その困難さの多くは組み込みシステムに起因する。このため、初学者はArduino上で動作するプログラムをいきなり開発するのではなく、本研究で開発したPC上で動作するプログラムの開発を十分行なった後に、Arduinoに挑戦したほうが円滑に学習できると考える。また、Arduino上の開発に挑戦し思うような成果が挙げられなかった場合にも、PCからLED照明を制御するプログラムは開発できるため、学習者の能力に応じたプログラミング経験を与えることができると考える。

4. サンプルプログラム

4.1 サンプルプログラム中のライブラリ機能 本照明機器を用いたソフトウェア開発を学生

が容易に行えるようにするため、サンプルプログラムを用意した。サンプルプログラム中には以下に示すライブラリ機能がある。

(a) 照明部制御の簡略化機能

本照明機器のLED制御ICに命令を送るには、本来は複数の命令を記述する必要があるが、これを `setLED(id,r,g,b);` と書くだけで制御できるようにすることで、細かい仕様を把握していない学生でも照明を制御できるようにした。なお、`id` は照明部の番号 (0～9)、`r`、`g`、`b` はそれぞれ赤緑青の強さ (0～255) を示す。

(b) 独自のsin関数機能の提供

実装上の工夫として、独自のsin関数 (`sin_deg`) を記述した。照明色を周期的に変化させるためにsin関数を用いると便利である。Arduino 言語で標準のsin関数は、テイラー展開により実装されており、PCに比べ格段に処理能力の低いArduinoでは実行時間がかかり、十分な速度で色を変化させる上で障害となりうる。そこで、1度単位のsin関数値をあらかじめ求めてテーブル(配列)に格納しておき、その値を用いることで関数値を得るようにし、十分な速度を実現した。なお、PC上で動作する

プログラムの開発環境では、テイラー展開で十分な速度が達成できるため、Processingに標準で用意されているsin関数を用いた。

(c) プログラム記述方法の統一化

上記 (a)、(b) の機能は、図2で示した5つの利用形態のいずれであっても同じように利用できるように統一した。しかし、それでもプログラムをそのまま記述すると、Arduino言語とProcessingの仕様の違いにより、プログラムコードに若干の差異が生じる。

その差異の主なものを図9に示す。まず、動作する場合の処理を記述するためのメソッドがArduinoでは`loop()`、Processingでは`draw()`という違いがある。そして、Arduinoでは`loop()`内に時間待ち処理を入れて処理の頻度を調整する必要があるが、Processingでは`draw()`内に時間待ち処理を記述してはいけないという違いがある。また、センサ値を取得するために使うアナログ入力の方法は、Processingでは`arduino.analogRead(0)`といった具合にオブジェクトを示す変数を指定する必要がある。

このため、新たに`loopLED`メソッドを設け、LED照明機器を制御するプログラムは必ず`loopLED`メソッド内に書くようにすることで

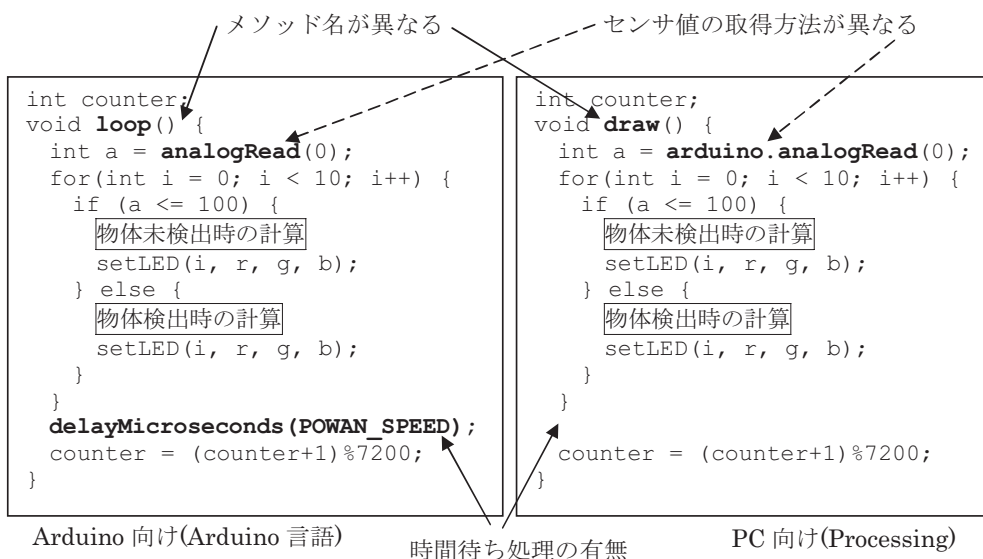


図9 Arduino向けとPC向けのプログラムの差異

2つの開発環境の差異を吸収した。LoopLEDメソッドの一例を図10に示す。このように統一化を図ったため、Processingで記述したプログラムのloopLEDの中身をArduinoのプログラムのloopLED内にコピーするだけでそのまま動作するようになり、ProcessingからArduinoに移行する場合の連続性を持たせることができた。ただし、この連続性は完全なものではなく、Processing-Arduino間の差異（言語の違い、ライブラリの違い等）が原因でそのままでは動かない場合もある。例えば、配列を使う場合、JavaとC言語の違いによって配列の初期化の記述方法が異なるため、修正する必要がある。しかし、統一化によって多くの差異が解消されたため、手直しが必要となる場合を大幅に減らすことができた。少なくとも、以下で述べる本教材のサンプルプログラムは全て、loopLED内を全く手直しせずにProcessingでもArduinoでも動作する。

4.2 プログラム例

以上の(a)~(c)のライブラリ機能を用いたLED照明のプログラム例を数種類作成した。図10にその一例を示す。この例は、学生がプログラミングを行う場合に、修正すべき場所のみ抜粋してある。つまり、このloopLEDメソッドを書き換えるだけで、学生は自由に作品を作ることが可能である。

この例は、距離センサの前に障害物がない場合は、照明が虹色に変化し、障害物がある場合は、白色に変化するプログラムである。なお、analogRead(0)は、Arduinoのアナログ入力ポート0につないだセンサの値(0~1023)が得る関数である。また、プログラム中のcounterは、loopLEDを呼び出す毎に1増えるカウンタ変数で、プログラムの別の場所に記述されており、学生は宣言や修正をする必要がない。

また、文献[1]で述べた複雑な色の変化をするデモ用作品も、図2に示した5つの動作環境の全てで動作するように改良した。

なお、本教材では学生の能力と興味に応じ

```
void loopLED() {
  int a = analogRead(0); // 100:遠い 400:近い
  for(int i = 0; i < 10; i++) {
    if (a <= 100) { // 距離センサが未検出の場合
      int phase = i*30+counter; // 1個ごとに色相が30度回転
      int r = (sin_deg(phase)+1)*128;
      int g = (sin_deg(phase+120)+1)*128;
      int b = (sin_deg(phase+240)+1)*128;
      setLED(i, r, g, b);
    } else { // 距離センサが検出した場合
      setLED(i, 255, 255, 255); // 白色で発光
    }
  }
}
```

図10 LED照明のプログラム例

た、以下の4つのレベルをプログラム例として用意している。

レベルA：PC上で動作するプログラムを書くことで照明作品を作る。

レベルB：PC上で動作するプログラムに近い感覚でArduino上で動作する照明作品を作る。

レベルC：Arduinoの入出力ピン、処理速度、メモリ容量など組み込みシステム特有の事項を考慮して開発する。

レベルD：プロセッサや周辺ICの仕様や制御方法、電子回路の理解といった、ハードウェアに密接し、より本格的な組み込みソフトウェア開発について学習する。

レベルAは、本論文で新たに設けたレベルであり、図2の(A)、(A')、(B)に相当する。レベルB~Dは図2の(C)、(C')に相当し、文献[1]の時点で設けたレベルである。

このように、本教材では4つのレベルを用意することで、PC上のプログラミングから、徐々に段階を踏んで本格的な組み込みシステム向け開発の経験が得られるように工夫している。

5. 類似した開発環境や機器との比較

5.1 フィジカルコンピューティング向け開発環境の比較

本教材は、フィジカルコンピューティングの一形態を実現したものと言える。フィジカルコンピューティングとは、物理世界とのやり取りを行なうコンピュータの利用形態で、多くの場合、PCと電子回路(I/Oモジュール、センサ、

アクチュエータ)を接続し、プログラミングを行なう[9]。本教材の場合は、センサとして距離センサを用い、アクチュエータとしてLEDを用いたフィジカルコンピューティングと言える。

現在フィジカルコンピューティングを行なうためのI/Oモジュールには主なものとして、Arduino、Gainer、Phidgetsがある[9]。これらのI/Oモジュールは、従来の組み込みシステム開発手段と比べ、開発が容易であるという特長がある。本教材で使用したArduinoとAVRマイコンやPICマイコンの開発環境との違いは、文献[1]で述べたため省略する。

Arduinoは手軽に使い低価格なことから、世界中に普及しており、様々な活用事例がある[2][3][10]。しかし、上記の各種I/OモジュールとLEDを用いた事例は数多くあるものの、ほとんどは小型のLEDを少量使用するものであり、本教材と同程度か上回る規模のLED照明機器は見つけることができなかった。また、Arduinoを利用した活用事例は、Arduino上で動作するプログラムの開発と、ProcessingからArduinoを制御するプログラムの開発のどちらかのみである場合がほとんどである。本教材のようにそれらの記述法を統一化して連続性を設ける試みは見つけることができなかった。

5. 2 照明機器の比較

本照明機器に類似したものに、市販の開発キットがある。しかし、開発キットの多くは発光色が一種類、多くて数種類であり、本照明機器の10種類より少ない。例えば、Texas Instruments社の開発キット[11]で本照明機器に類似した照明が実現できるが、PWM出力が14チャンネルのためRGBでは4種類の発光しかできず、Texas Instruments社独自のプロセッサと開発環境を用いるため開発が難しい。これに対し、本照明機器は30チャンネルを駆動し、制作費は上記開発キットの3分の1以下であり、Arduinoを用いたため開発が容易で、さらに開

発が容易なPC上のProcessingによるプログラミングも可能である。

光量が大きく、数多くの色を表現できる他の手段として、舞台照明機器がある。舞台照明機器では調光制御にDMX512-Aというプロトコルが普及しており、PCで制御するソフトも販売されている[12]。しかし、RGBカラーで明るい照明を実現しようとする高額になるという欠点がある。また、調光はあらかじめ決められたパターンから選択することしかできない場合が多く自由度が低い。DMX対応の舞台照明機器を用い、本照明機器と同等の照明を実現するための価格を試算し、本照明機器の制作費と比較したところ、本照明機器の費用はDMX機器の約3分の1であった。

6. おわりに

本研究では、学生のプログラミング能力を育成する一手段として、発光色をプログラムで制御できるLED照明機器を教材として開発した。個々の照明には距離センサが内蔵されているため、学生は、照明の前に人が立つなどすると、それに反応し照明の発光が変化する作品が開発できる。本教材ではPC上で動作するプログラムの開発環境、LED照明のエミュレータ、小型コンピュータ上で動作するプログラムの開発環境、および本格的なLED照明と簡易装置を組み合わせることで、5種類の開発方式を提供した。これにより、初学者向けの開発環境から始め、徐々に高度な開発環境を利用することが可能となり、円滑なプログラミングの学習が期待できる。

今後の予定としては以下の5つの項目について検討していく。

- 照明の形態として今回は10本の柱状としたが、これに限定されるものではない。様々な形状や天井設置、壁設置など、他の形態も製作し多様化するとともに、使用したLED制御ICは、連結して容易にチャンネル数を増やせるため、照明の本数をさらに

増やして表現力を向上させたい。

- 観賞用でない一般の照明機器も今後開発したい。一般用途であれば、白色LEDを使用し効率を上げることが可能である。各種センサにより、屋内にいるヒトの状況を認識して、省エネルギーだけでなく、仕事空間や生活空間の質の向上を狙いたい。
- 本教材では、ヒトの認識に距離センサを用いたが、Kinectなどの二次元距離センサや焦電センサなどの他のセンサを活用することで、より詳細なヒトの行動認識を用いたり、検出範囲を向上したい。これにより、学生が作ることができる作品の適用範囲を広げることが期待できる。
- 現状の本教材におけるプログラミング教育環境は不完全である。ProcessingとArduino自体の特徴により初学者にも敷居が低いものになっているが、ソフトウェアの教材としては数種のサンプルプログラムしか存在しない。今後、プログラムの例題とその解説を数多く用意し、教材を充実させたい。
- 本教材をゼミや授業で実際に適用する必要がある。その結果に基づいて、本教材の改善を図っていきたい。

【参考文献】

- [1] 大見嘉弘：組み込みシステム用教材の開発とLED照明への適用，東京情報大学研究論集 Vol.15, No.2, pp.83-95 (2012)
- [2] Banzi, M.: Getting Started with Arduino, O' Reilly (2009), 船田巧 (訳)：Arduinoをはじめよう，オライリー・ジャパン (2009)
- [3] Arduino Team: Arduino Uno, <http://arduino.cc/en/Main/arduinoBoardUno>
- [4] Arduino Tema: Arduino MEGA 2560, <http://arduino.cc/en/Main/ArduinoBoardMega2560>
- [5] Processing, <http://processing.org/>
- [6] Reas, C. and Fry, B.: Processing: A Programming Hand-book for Visual Designers and Artists, MIT Press (2007)
- [7] Main Page – Firmata, <http://firmata.org/>
- [8] Tronne, P: Why the Arduino Won and Why It's Here to Stay, <http://blog.makezine.com/archive/2011/02/why-the-arduino-won-and-why-its-here-to-stay.html>
- [9] 小林茂：フィジカルコンピューティング概論，情報処理 Vol.52, No.8, pp.914-917 (2011)
- [10] 菅野創：Arduino入門－15万台の販売実績を持つマイコンボード－，情報処理 Vol.52, No.8, pp.922-925 (2011)
- [11] Texas Instruments Inc.: マルチDC / DCカラー LED キット, <http://focus.ti.com/jp/docs/toolsw/folders/print/tmdsrgbledkit.html>
- [12] 藤井直：ステージ・舞台照明入門，リットーミュージック (2006)

