

組み込みシステム用教材の開発とLED照明への適用

大見 嘉弘*

ソフトウェア開発者を育成する大学の課程で用いる組み込みシステム用教材を開発し、LED照明機器に適用した。この照明は10本の柱状照明からなっており、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。発光色はRGB各12ビットと細かく指定でき、高速に変更できるため、滑らかな色の変化が表現できる。また観賞用の照明として十分な光量を確保しており、かつ拡散板により目に優しい発光を実現した。また、各照明に距離センサとマイクを内蔵しており、それらを入力とし反応する作品が製作可能である。本教材を用いて学生が照明作品を製作することで、学生のモチベーションを高め、創造的な姿勢の醸成、プログラミング能力の向上を図ることが期待できる。また、教材の難易度を3つに分けることで、学生の能力や興味に応じた学習が行えるよう工夫した。学生が本教材を用い短期間で作品を製作したことで、本教材の有効性が確認された。

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 for Embedded System and Application to LED Lighting

Yoshihiro OHMI*

A teaching material of embedded system is developed and is applied to a LED lighting device. This material is used in the course of software development. This device consists of ten pillar type lights, and each light color can be controlled from a microcomputer. Colors of these lights can be specified 12bit color each RGB channel, and it can be changed quickly. In addition, this device has practical power of light for ornament and the light is mild for eyes because of optical diffuser. Each light module has a distance sensor and a microphone, and some works which response to these devices can be developed. For developing light works by students, it is expected to improve educational motivation, to raise creating status and to improve skill of programming. Students can learn which is suitable to their skills and interests, because this material is composed three difficulty level. Also, validity of this material is confirmed, because some students developed works in short duration using this material.

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

1. はじめに

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は二の次という場合が多い。例えば、何らかの処理を行い、出力は標準出力としてテキストで表現、そして、入力がある場合はキーボードからテキストを打ち込む場合が多い。今日、コンピュータユーザが触れるアプリケーションソフトウェアの大多数はグラフィカルユーザインタフェース（GUI）を用いており、テキストのみを入出力としたプログラミング課題を専らこなしていると、勉学のモチベーションを保ちづらいという傾向が見受けられる。しかし、一般にGUIプログラミングは複雑で習得が難しい傾向にあり、初学者に対しては入出力をテキストデータに限った課題のみにせざるを得ないという事情がある。さらに、今日ではGUIに限定されず、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等で周辺の実世界を捉え、その情報にPCで生成された情報を重畳する仮想現実などである。

このように、キーボード、マウス、ディスプレイを入出力とする旧来のPCのインタフェースに限らないコンピュータインタフェースが身近になりつつある現在、旧来のインタフェースを用いたプログラミング教育のみでは、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。今後、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に従事する技術者が今後増加すると予想され、学生が将来のシステム開発の職業像を捉えたり、必要な能力を身につけるためにも、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と考える。

そこで、このような新しいインタフェースの

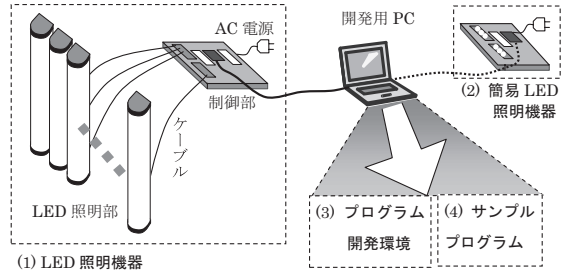


図1 LED照明教材の構成

ひとつとして、小型コンピュータで制御できる本格的なLED照明機器を教材として開発した。本教材は、以下の4つから構成される（図1）。

(1) LED照明機器

センサを内蔵しており、人間の行動を検出できる。また、理論上36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ことを評価実験で確認した。

(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おり、プログラミングが苦手な学生でもプログラム作品の作成が可能となる。

(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。

本教材を用いることで、ハードウェアに密接なソフトウェア開発が行えるため、組み込みシステムにおけるソフトウェア開発の導入部の役割を果たすことを目標としている。また、本教材は学生の能力や興味に応じて3つのレベルで学習が行える。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、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（左側）に、実際の照明部を図2に示す。照明は10本の柱状のLED照明であり、これらは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、RGBごとに12bit、計36bitのフルカラーで色を指定することができる。それぞれのLED照明はケーブ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、動作させることができる。さらに、それぞれのLED照明には、距離センサとマイクロフォンが内蔵されており、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。

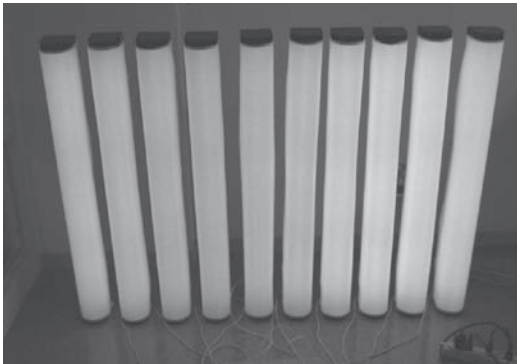


図2 LED照明機器の照明部の写真

(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)

色の変化を楽しむことを想定して床に置くことにした。日本古来の行灯（あんどん）に類似した形態である。

LED自体にはテープ状のRGB LEDアレイを用いた。1本の照明部に長さ85cmのLEDアレイを使用しており、51個のRGBチップLEDから構成されている¹⁾。図4にLEDアレイの拡大写真を示す。個々のチップLED内には、赤、青、緑の3つのLEDが配置されている。また、電流制限抵抗が内蔵されているため、12Vを印加するだけで発光する。開発した照明では、パルス幅変調（PWM）によって高速に点滅させることで、調光を行い、様々な色を発光する。

照明部は、表面に拡散板を設けることでLEDの光を拡散している。ほとんどのLEDは点光源であり、そのまま使用すると照明を直視した場合に目に焼き付き、まぶしい恐れがある。開発した照明は直接目で見て楽しむことを主目的としているため、光を拡散して「柔ら

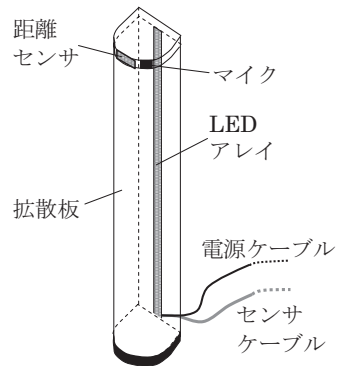
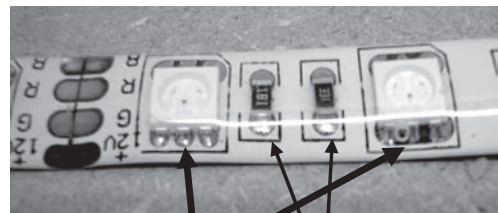


図3 照明部の模式図

2.2 照明部の設計と実装

照明部は実際に照明として光る部分である。照明部の1本は、高さ95cm、幅9cm、奥行き10cmの柱状である。模式図を図3に示す。これを10本、床の自由な場所に配置して使用することを想定している。これらは1本ずつ異なった色で発光することができる。一般に照明は天井に設置することが多いが、今回は人が近寄って



RGB チップLED 電流制限抵抗

図4 LEDアレイの拡大写真

かい光」を表現するようにした。また、RGB LEDは1チップでR、G、Bが発光できるが、それぞれの位置がずれているため、それらの光をうまく混じり合わせるために光の拡散が必要であった。なお、拡散板には、入手性や価格を重視して薄手の障子紙を用いた。

照明部の最上部にはセンサが内蔵されている。センサは距離センサとマイクであり、地上から95cmの高さに設置している。したがって、成人の腹から胸辺りの高さとなる。距離センサは、赤外線を用いた測距モジュール（SHARP製GP2Y0A21YK）を用いている。このセンサは、複数を近接した場合の混信が生じにくいという利点があり、照明部を近くに並べても機能する。しかし、対象物が80cm程度より遠い場合は検知できないという欠点がある。マイクは、シリコンマイクを使用したため、小型で、十分に増幅した信号を得られノイズに強いという特長がある。これらは、4端子のセンサ用ケーブルを介して、制御部に接続する。

2.3 制御部の設計と実装

(a) 制御部の概要

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5、回路を図6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[1]。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の難易度が低いのが特徴である。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、2.5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[2]。なお、Arduino Unoのポート数で不足する用途には、ポート数の多い機種であるArduino MEGA2560を用いることを想定している[3]。

その他のArduinoの特徴としてシールドの存在がある。Arduinoでは、Arduino本体の上に

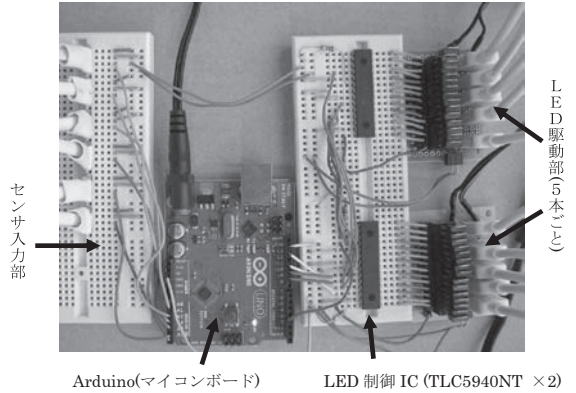


図5 制御部の写真

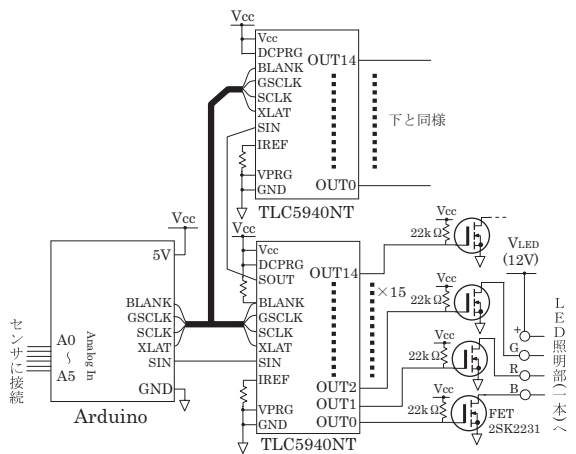


図6 制御部の回路図

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: 5V
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: 2KB
EEPROM容量: 1KB

シールドと呼ばれる基盤を挿し込むだけで機能拡張が行える。シールドは挿し込むだけで機能するため配線作業の必要がない。これは、Arduinoの入出力ピンの配置が規格化されてい

るためである。シールドは様々な種類のものが販売されている。例えばイーサネットシールドを挿すと Arduino を LAN に接続でき、GPS シールドを挿すと GPS 位置情報を利用できる。

(b) LED の PWM 制御と駆動方法

本照明機器では10本の照明部を制御するため、30本のPWM出力が必要である。しかし、Arduino Unoには6本のPWM出力ポートしかないため、さらにLED駆動専用IC (Texas Instruments TLC5940) を使用した[4]。TLC5940は、PWM出力を16本持ったICであり、それぞれの出力で4096階調 (12bit) の調整が可能である。RGB出力に用いると36ビットとなり、理論的には約687億色の発色が可能となる。本照明機器では、TLC5940を2個使用することで30本のPWM出力を実現している。また、TLC5940は、制御するための通信レートが30MHzと高いため、30チャンネルの出力であっても、ヒトの視覚特性 (あらゆる条件において60分の1秒より高速な点滅は知覚できない[5]) に対して十分に高速で滑らかな色変化が実現できるためである。

しかしながら、TLC5940は本来は出力ポートにLEDを直接接続するが、出力ポートの定格電流は120mAであり、照明部のLEDアレイの想定電流500mAを駆動することができない。このため、駆動回路を別途用意した。本照明機器では、駆動回路にパワー MOSFET (Metal Oxide Semiconductor Field Effect Transistor) (東芝 2SK2231) を使用した。その理由は高速に大電流をスイッチングする手段として従来はパワートランジスタやサイリスタ等が用いられてきたが、それらに比べパワー MOSFETは回路設計が容易なため近年主流になりつつあるためである。しかし、TLC5940の出力は、LED点灯時0V、消灯時ハイインピーダンスであるため、そのままFETのゲートに接続してスイッチングすることができない。そこで図6に示すようにFETのゲートを抵抗でプルアップすることでスイッチングを実現した。この場合、点

灯と消灯の状態が反転してしまうが、FET 1個当たり抵抗1個の付随部品だけで構成でき回路が簡単になる。点灯と消灯が反転する仕様については、Arduino上のプログラムで反転して対応することにする。

(c) センサ入力

本照明機器の照明部に内蔵したセンサは、Arduinoに接続して使用する。距離センサについては、距離に応じた電圧が出力されるので、これを直接Arduinoのアナログ入力ポートに接続する。Arduinoではその入力がAD変換され10bitの値として得られる。マイクについては、増幅、レベル変換の回路を介してArduinoに接続する必要がある。

なお、何個のセンサを利用するかは、どのような作品を製作するかによって決まる。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を使用する。

2.4 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1m、幅1m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、7)。これは、ブレッドボード上にRGB LEDを10個並べたもので、LED1個が照明部1本に相当する。LED1個の輝度は大きくないため、激しいちらつきを起こしても視覚的な悪影響は小さい。また、部品点数が少ないため、製作も容易である。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と考える。

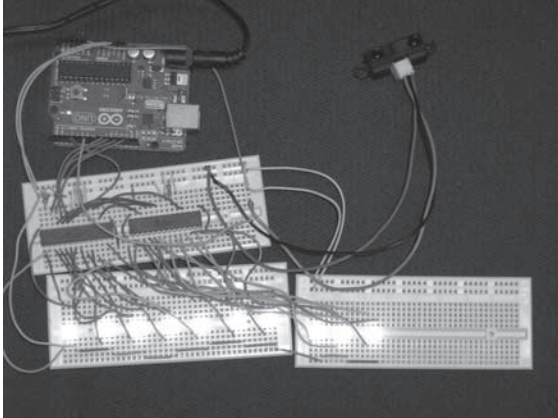


図7 簡易LED照明機器の写真

2.5 ソフトウェア開発環境

本教材のソフトウェア開発にはArduinoの標準的な開発環境 (Arduino IDE) を用いる。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[1][6]。Processingは平易にソフトウェア開発が行えるよう工夫された開発環境であり、Arduino IDEもその特徴を引き継いでいる。このため従来の組み込みシステム開発と比べ格段に開発が容易に行えるという特長がある。本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おり、約3割の学生が履修しているため、より円滑にArduino IDEを用いたソフトウェア開発が可能となっている。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。しかし、Arduino言語は極力Processingに類似するように工夫されている。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[1][7]。

図8に開発に使用するArduino IDEの画面を示す。単一のエリアにプログラムコードを打ち込み、PCとArduinoをUSB接続し、上部にあるアップロードボタンを押すだけで、コンパイル→リンク→Arduinoへのアップロード→Arduino上での実行が行える。従来の組み込みシステム開発では、テキストエディタでコードを作成し、コンパイラ&リンカがバイナリコードを生成、それをライティングソフトで対象機器のFlashROMに書き込み、実行するといった煩雑な操作が必要が多かった。最近ではIDEの主流であるEclipseを用いて開発を行う組み込み機器が増えているが、Eclipseはウィンドウ内が数多くのフレームで区切られ、メニュー項目やコマンドが多岐にわたり、特に初心者には難しいが、Arduino IDEは格段にシ



図8 Arduino IDEの画面

ンプルで分かりやすい。

3. 本照明機器の教材としての適用

3.1 教材として独自に用意した機能

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、以下の機能を用意した。

(a) 照明部制御の簡略化機能

本照明機器のLED制御ICに命令を送るために、Arduino用のライブラリである `tlc5940arduino` を使用した[8]。このライブラリを直接使用して照明部の制御を行う場合、図9のように記述する必要がある。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(0~4095)を示す。このように、LED駆動ICの出力ポートの番号と各照明の番号との対応を把握したり、送出するデータを反転すること、命令を送るために`update`コマンドの実行が必要なことを把握しないとコーディングできない。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。

```
int ch = (id/5)*16+(id%5)*3;
Tlc.set(ch,4095-g);
Tlc.set(ch+1,4095-r);
Tlc.set(ch+2,4095-b);
Tlc.update();
```

図9 プログラム例

(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。照明表現で求められるsin関数の精度は1度単位で十分であり、問題は生じない。しかし、より細かい精度を実現するためテーブルを大きくしたところプログラムが異常動作した。Arduinoのメモリは2KBしかなく、float型配列の場合、たった512の要素で満杯となることが原因であった。

(c) サンプルプログラム

以上の(a)、(b)の機能を用いたLED照明のプログラム例を作成した(図10)。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、この箇所のみ書き換えるだけで、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言語に標準で用意されている関数で、Arduinoのアナログ入力ポート0につないだセンサの値(0~1023)が得られる。

```
int counter;
void loop() {
  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)*MAXB/2;
      int g = (sin_deg(phase+120)+1)*MAXB/2;
      int b = (sin_deg(phase+240)+1)*MAXB/2;
      setLED(i, r, g, b);
    } else { // 距離センサが検出した場合
      setLED(i, 4095, 4095, 4095); // 白色で発光
    }
  }
  delayMicroseconds(POWAN_SPEED); // 時間待ち
  counter = (counter+1)%7200;
}
```

図10 LED照明のプログラム例

さらに、本照明機器を用いたデモ用の作品を製作した。これは、10本の照明部がそれぞれ色相を変えて純色で発光するもので、平たく言えば虹のような光の表現を行う。色相は左右に複雑に変化するように工夫した。図11に色相の変化を示す。この複雑な色相変化を実現するために、以下の数式を用いた。

$$\theta = t + 100\sin(3t + \sin 0.4t)$$

※ θ とsinは度単位、 t は経過時間

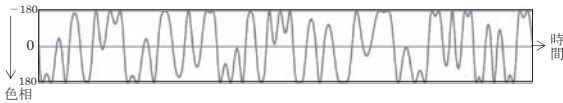


図11 作品の色相の変化

また、照明部の前にヒトが立つと虹色から白色のみの発光に変化するようにした。ヒトの検出には照明部に内蔵された距離センサを使用した。虹色から白色への変化は徐々に変化するようにし、ヒトが遠くにいる場合はゆっくりと、近くにいる場合は素早く変化するようにした。

このように、複雑な色変化とセンサによる明瞭な反応を実現したプログラムを学生へのサンプル例題として用意することで、学生がプログラム作品を製作するうえで参考になるようにした。このプログラムは、43行（実質的な部分のみ。コメント、空行を除く）であり、表現の複雑さを鑑みると短いもので、例題として適していると考ええる。

3.2 レベル別教材の提供

本教材では、学生の能力と興味に応じた、以下の3つのレベルの教材を提供している。

レベルA：PC上で動作するプログラムとほぼ同じ感覚で作品を開発する。

レベルB：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C：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3.1節 図10に示すような、ほとんど組み込みシステムであることを意識せずにプログラム開発が行えるレベルである。ここでは、独自に開発したsetLED, sin_degとArduino言語標準のanalogInputの各関数が制御方法の詳細を隠蔽しているため、それぞれの関数の使用方法さえ理解していればハードウェアの知識がなくても作品の製作が行える。このレベルであれば、本学情報システム学科の

学生の多くが作品を製作可能である。

そして、組み込みシステム開発に興味があり、より詳細について学習したい学生はレベルBの教材を用いて学習が行える。レベルBでは、setLEDの実装を学ぶことでLED制御ICの出力制御の方法をライブラリを利用するレベルで理解することができる。また、sin_degの実装を学ぶことで、組み込み向けプロセッサの処理速度の低さやメモリの少なさに対応する方法について学習することができる。

さらに本格的に組み込みシステムの技術を学びたい学生はレベルCの教材を用いて学習が行える。レベルCでは、Arduino言語のソースコードやArduinoのプロセッサであるAVRマイコンの解説書や仕様書、LED制御ICを用いるライブラリのソースコードやICの仕様書が教材となる。Arduino言語や周辺のライブラリは全てオープンソースであるため、このようなことが可能になっている。

4. 本教材の評価

4.1 光量の評価

本照明機器の照明部が実用的な明るさを実現しているか確認するために、光量を測定した。測定では、10本の照明部を密着させ一列に並べ、その中央から正面に100cm離れた位置における照度を測定した²⁾。10本全てを最大発光させた場合263 lx、中央の1本の場合32 lxであった。学校の教室内の照度は300 lx以上、500 lx以上が望ましいと定められている[13]。したがって、教室を照らす照明としては暗く、周囲を薄暗く照らす程度である。なお、本照明機器のような観賞用照明や類似の舞台用照明については、JIS等において光量の基準が定められていないが[5]、教室と同等な明るさにするとまぶしすぎると推測する。本照明機器は鑑賞用照明であり、照明を直接見ることを前提としている。例えば、教室照明を直接目視すると目に焼き付きを生じ、まぶしいと感じるため、やや暗いほうが適当と考える。

上述したように直視する照明の光量に関する基準がないため、評価実験を行い明るさの評価を行った。評価実験では、本照明機器の明るさをどう感じるかを5段階評価で回答させた。被験者は大学生17名であった。その結果を図12に示す。光は、6章で述べた作品で虹色に変化するものと白色で最大発光したものの2種類について回答させた。なお、周辺照度は本学のオープンキャンパスにおいて展示した場合とほぼ同じ状態で実験を行った（LED照明消灯時の周辺照度150 lx）。その結果、虹色は平均として普通よりややまぶしい（5段階評価で3.29）、白色は平均として普通よりやや暗い（5段階評価で2.88）という結果となった。以上の結果から、本照明機器は十分に実用となる光量を発していると判断する。

なお、現状の光量で不十分な場合には、使用するLEDの数量を増やすことで容易に対応可能である。駆動部のFETの駆動能力は余裕があり、照明部1本当たりのLEDを今の3倍程度に増やしても対応可能である。

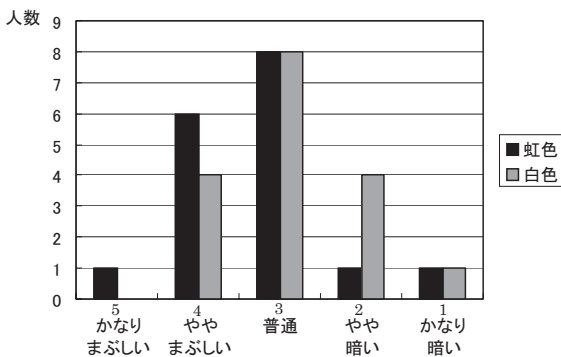


図12 光量の評価実験の結果

4.2 処理速度の評価

本照明機器は、十分な速度で発光色を変化させることで多彩で滑らかな色表現を行えることを目標としている。そこで処理速度を評価するために、発光色の変更にかかる時間を測定した。ヒトの視覚特性を考慮すると1秒間に60回以上、余裕を見て100回以上変更できることが

望ましい[5]。10本全ての発光色を変更するために必要な時間は、139 μ sであった。つまり1秒間に7,000回以上変更できる速度であり、十分であることが分かった。その理由はLED制御IC (TLC5940) の通信レートの高さである。ただし、実際にはArduinoの処理能力が高くないため、処理速度について注意する必要がある。例えば3.1節で述べたデモ作品は、sin関数を高速化したにもかかわらず複雑な処理のせいで、発光色の値を求める処理時間が7.4ms（1秒間に135回）かかってしまった。1秒間に100回以上処理しており、上記の基準を満たしているが、これ以上複雑な処理をすると基準を満たさなくなる恐れがある。

4.3 ソフトウェア開発環境とサンプルプログラムの評価

本教材を用いて、著者のゼミ生3名が大学祭向けに2つのプログラム作品（作品1、2）を製作した。学生は、本教材のサンプルプログラムを参考にしながらソフトウェア開発環境で開発を進めた。作品1は、照明部の距離センサを3つ使い、それぞれ赤、緑、青の光の強さを距離に応じて変化させ様々な色の光を発するものである。作品2は、照明部の距離センサに手などを近づけると距離に応じて音程が変化する楽器である。また、音が出ると同時に、照明はオーディオのレベルメータ風に色が変化する。なお、楽器の音を出すために楽器シールド[14]をArduinoに挿して使用した。Arduinoを用いたため、このように必要な機能があればシールドを用いて容易に機能を追加できた。作品1、2の制作期間はそれぞれ約2週間であった。なお、作品1を製作した学生は、今回初めてArduinoに触れ、3.2節で述べたレベルAの教材のみを用いた。作品2を製作した学生は、半年前からArduinoについて学習を始めており、今回はレベルAに加え、部分的にレベルBの教材を用いた。setLEDの実装を理解し、楽器シールドの使用方法について学んだためである。このように、短期間で学生が作品を完成したことが

ら、本教材の有効性が確認できた。

5. 類似した開発環境や機器との比較

5.1 組み込みシステム開発環境の比較

本教材で用いているArduinoに類似した開発環境との比較を行う。Arduinoと性能、価格が近く、幅広く用いられているプロセッサとして、Atmel社のAVRマイコン[9]とMicrochip社のPICマイコン[10]の開発環境を取り上げる。

(a) ArduinoとAVRとの比較

AVRマイコンはArduinoに採用されているプロセッサである。Arduinoでは、AVRマイコンの一品目であるATmegaシリーズが採用されている。ここでは、Arduinoの開発環境とC言語によるAVRの開発で広く用いられているWinAVRを用いた開発を比較する。まず、コーディングの難易度について比較する。AVRは、同レンジの8 bitマイコンの中では、レジスタ数が豊富で、命令セット体系やアーキテクチャに直交性が高く、C言語による開発を意識したものになっている。このため、PCにおいてUNIXやMac OS Xの標準Cコンパイラであるgcc (GNU C Compiler) が用意されているほどである。なお、WinAVRではCコンパイラとしてgccが用いられる。しかし、gccを用いても、AVRのハードウェアの詳細を把握していないと開発できない。例えば、入出力ピンの仕様を把握する必要がある。AVRに限らず、組み込み向けプロセッサは、少ないピン数で多くの機能を提供する目的で、1つのピンに複数の機能を持たせ、機能を選択して利用することが多い。例えば、Arduino UnoのプロセッサであるATMega328は、入出力ピンにPB0~7、PC0~6、PD0~7といった名前が付いており、例えばPB0はPB0 (PCINT0/CLKO/ICP1) と複雑に表記される。これはPB0が、PCINT0、CLKO、ICP1の各機能のうちの1つを選択して使用できることを表している。これに対して、ArduinoはこれらのピンをDIGITAL 0~13、ANALOG IN 0~5と表記している。この名称はピンの機能を

率直かつ統一的に表しており、組み込みシステム開発に馴染みのない者であっても理解しやすいものになっている。また、AVRのこれらのピンはポート (PB、PC、PD、…) 単位でデータをまとめて扱う。例えば、PB7の入力値 (HIGH/LOW) を得る場合、PB0~7の計8ビットをまとめて得てから、80 (16進数) との論理積を求めることになる。このようにAVRでは入出力ポートの仕組みやビット演算の知識がないと開発できないが、Arduinoは、その詳細をライブラリで吸収しており、ピン番号を指定するだけ (上記例の場合digitalRead(7)) で入出力が行える。

次に、開発手順の手間について比較する。AVRの開発環境としてWinAVRを用いる場合、典型的な開発手順は以下のようになる。(1) AVRライター機能のある実装基盤をPCに接続する。(2) WinAVRを起動しプロジェクトを新規作成する。(3) Makefileを記述する。(4) C言語でコーディングする。(5) Make Allを実行するとhexファイルが生成される。(6) 書き込みコマンドを実行しhexファイルをAVRに書き込む。(7) AVRマイコンをリセットしてプログラムを実行する。その後のデバッグ作業は(4)~(7)を繰り返すことになる。これに対して、Arduinoでは、以下の手順となる。(1) ArduinoをPCに接続する。(2) Arduino IDEを起動し、コーディングする。(3) Arduino IDEのUploadボタンを押すと、実行コードがArduinoに転送され、自動的にプログラムが実行される。その後のデバック作業は、(2)、(3)を繰り返すだけであり、格段に手間が少ない。

(b) ArduinoとPICとの比較

まずコーディングの難易度を比較する。PICはAVRと競合する8 bitマイコン³⁾であり、特に日本では参考書が多いため広く採用されている。PICはピンの入出力などの設定を変更するための特殊レジスタが数多くあり、設定にバンク切り替えを必要となることが多い。また、ス

タックが8段しかなく、一度に利用できるメモリ空間が狭いためページングを多用する。このように癖の強い仕様であるため、Cコンパイラが生成するコードの実行効率が悪く、C言語による開発に向いていないプロセッサと言える。このため、現在でも開発の主流はアセンブラ言語である。また、C言語で開発する場合も、アセンブラ言語による処理をそのまま置き換えただけの組み込み関数を呼び出す必要があり、アセンブラ言語の知識なしでは開発が困難である。

次に開発手順の手間について比較する。PICの開発手順は、上記のAVRより手間がかかる傾向にある。PICに実行コード等を書き込むライターに必要な部品や供給電圧の高さから、実装基盤にライター機能を設ける場合、AVRより実装が面倒で費用がかさむ。このため、実装基盤にライター機能を設けない場合が多いが、デバッグのたびにPICマイコンをライターと実装基盤の間で差し替えるという煩雑な手間が生じ、開発効率が悪い。

以上、AVRとPICによる開発との比較を行った。従来の組み込みシステム開発では、ハードウェアの専門知識を持つ者のみが開発できるという傾向があったが、Arduinoの登場で、専門知識を持たないデザイナー、アーティストなどが、電子回路を用いたプログラム作品を手軽に作成するようになってきている[1]。

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、各チャンネルの駆動力は5Wである。さらにTexas Instruments社独自のプロセッサと開発環境を用いるため開発の難易度が高い。これに対し本照明機器は30チャンネルを駆動し、

各チャンネルの駆動力は20Wで、駆動部の制作費は上記開発キットの3分の1以下であり、Arduinoを用いたため開発難易度が低い。

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。これらは舞台照明だけでなく、メディアアートやインタラクティブアートにおいても用いられている。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⁴⁾ 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⁵⁾ 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。本照明機器は4.1節で述べたように観賞用の照明として必要十分な光量があり、かつ、10本の照明を別々の色で発光でき、安価である。そして、このように本格的な照明を用いたプログラミング教材は、他に類のないものである。

6. おわりに

本研究では、観賞用照明として十分な光量があり、発光色を小型コンピュータで制御できるLED照明機器を教材として開発した。本教材のプログラム開発環境として、(A) 組み込みシステムということをはほとんど意識せずに開発できる、(B) プロセッサや周辺ICの性質をある程度理解して開発する、(C) ハードウェアの詳細を理解して開発する、という3つのレベルで学習できるようにし、各学生の能力や興味に応じた教材が提供できるよう工夫した。また、学生が本教材を用い短期間で作品を製作したことで、本教材の有効性が確認された。

今後の予定としては以下の4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。想定している照明機器は以下の通りである。オフィス等の多数の者が出入りする屋内にセンサとLED照明を設置し、屋内にいる人の状況に応じて適切な照明を提供する。例えば、机に距離センサ、あるいは天井にカメラを設置して人の状況を検知し、天井と机の近辺に設置したLED照明を調光する。まず最初は、人がいる場所のみ照明を付けるという単純な方式となるが、ゆくゆくは例えば、人が仕事をしている時とくつろいでいる時などの状況を検知して、適切な明るさで照らすシステムを実現したい。これを実現することで、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目標んでいる。
- 照明部に内蔵したセンサについて、距離センサを用いたプログラムは容易に開発できるが、マイクは現状では困難である。Arduinoのアナログ入力に接続できる回路を開発する必要があり、さらに接続できてもArduinoの処理能力の低さにより、瞬発的な音を取りこぼす恐れがある。今後は対応する回路の開発を行いたい。
- 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。Arduino自体の特徴により初心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。

【注】

- 1) チップ当たりの輝度は赤0.75cd 緑1.12cd 青0.4cdである。
- 2) 使用した照度計Tondaj LX-1010Bは簡易測定器であり、無視できない誤差がありうる（仕様としては精度 $\pm 5\%$ ）。また、照明器具の光量を示すには、光束や光度や輝度の値を示すのが一般的であるが、測定が大掛かりで高価であるため、照度のみを測定した。
- 3) 16bit版のPICもあるが、高価で開発環境が大掛かりであるため、本稿では8bitミッドレンジ版であるPIC16Fシリーズのみと比較する。
- 4) 例えば、高機能なDMXコントローラの一つであるElation Show Designer 1であっても、事前に設定したパターンから選択するか、フェーダー等を操作してリアルタイムに調光することしかできない。
- 5) この試算は市販製品の中で特に安価な機種を選んだ結果である。なお、この試算の構成では制御にPCを必要とするがPCの価格は含んでいない。また、PC上の制御ソフトウェアには、事前に設定した照明パターンから選択する機能しかなく、プログラミング機能は提供されていない。

【参考文献】

- [1] 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)
- [2] Arduino Team: Arduino Uno, <http://arduino.cc/en/Main/arduinoBoardUno>
- [3] Arduino Tema: Arduino MEGA 2560, <http://arduino.cc/en/Main/ArduinoBoardMega2560>
- [4] Texas Instruments Inc.: TLC5940NT, <http://www.ti.com/product/tlc5940>
- [5] 照明学会編：照明ハンドブック, オーム社, (2006)
- [6] Reas, C. and Fry, B.: Processing: A Programming Handbook for Visual Designers and Artists, MIT Press (2007)
- [7] 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>
- [8] Leone, A.: tcl5940arduino, <http://code.google.com/>

p/tlc5940arduino/

- [9] 山根彰：AVRマイコン・リファレンス・ブック，CQ出版（2006）
- [10] 後閑哲也：C言語によるPICプログラミング入門，技術評論社（2009）
- [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）
- [13] 文部科学省：学校環境衛生基準，文部科学省（2009）
- [14] SparkFun Electronics: Music Instrument Shield, <http://www.sparkfun.com/products/10587>

