

論文

iPhoneOS と Arduino を用いた 照明制御システム

小山 祐 諏訪 敬祐

本研究は既存の「Arduino を用いた照明制御システム」においてマイコンとスマートフォンの組み合わせによる新たな可能性に着目し、従来システムの改良を行うことで、さらなる消費電力量の削減を目的とする。iPhone と Arduino のプログラムの変更を行い、4段階の輝度制御を実現し、昼間等明るさが最大である必要がない場合の輝度調整が可能で、内蔵時計の機能を ON にすることで意識しなくても明るさを制限することにより省エネを実現できることを実証した。

キーワード：iPhone, Arduino, 照明制御, Objective-C, LED

1 はじめに

1.1 研究の背景

近年、携帯電話と超小型パソコンとしての PDA を統合したものと言われているスマートフォンが、急成長を遂げている。スマートフォンは 1990 年代末から 2000 年代前半にかけて誕生し、その際主導権を握ったのは Symbian と BlackBerry であった。しかし、当時、日本では、多機能携帯電話が独自の発展を遂げており、スマートフォンへの関心は低かった。その後 2000 年代後半になり、スマートフォンのオープンな環境での利用のしやすさや、iPhone3GS の発表とソフトバンク社の積極的な宣伝やキャンペーンにより、スマートフォンへの関心が日本国内で高まり、2009 年に爆発的に普及し、さらに 2010 年にはスマートフォン OS 「Android」の搭載機が多く発表され、大きな国内市場が確立した。現在の利用率は 22% まで達し、2010 年 9 月時点の 9%、2011 年 4 月時点の 14.8% の数値と比較すると大きな拡大を遂げていることがわかる^[1]。

スマートフォンの中でも iPhone が各種センサーやタッチセンサーの反応が、現状他のスマートフォンと比べ優れているため、照明制御に最適であると考え iPhone を利用することとした。

現在は、感性の時代、多様化の時代となってきた背景から、好ましいといわれる状態は多岐にわたり、居住空間の利用はそれぞれの欲求に応じ、専門的、個性的になっている。

一つの空間における多目的使用に伴い、照明の利用も使用用途に合わせ、調光しなければならない。さらには、使用空間の時間的変化に合わせ、照明の演出を変えることが好まれている。例えば、喫茶店のように、昼間と夜間で業務形態や客層が変わり、それに対して照明演出を変えなくてはならない店舗がある。また、目的場所のみの照明の利用と全設備点灯での電力消費は前者のほうが少なく省エネに貢献することは、明白である。

さらに、昨今の電力供給問題に対して、照明に対する考え方が変わり、明るすぎる照明から適切な明瞭による照明の利用が重要視されてきている。節電のためにオフィス照明の間引きを行うことは、間引きを行わずに作業環境が悪化することになる。

さらに昼間照明としての問題点として、以下の図 1 のように、日光によって影になる部分や、日光利用によって人工照明の明瞭を減光することが可能な照明等という問題がある。また、壁側から窓側の人を見たときに、日光のみだと窓側の人の顔が暗くみえてしまう。

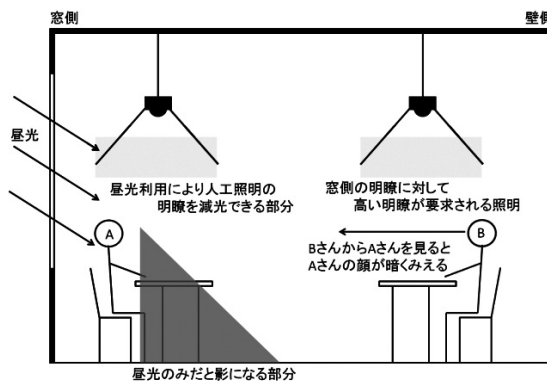


図 1 昼間における室内照明の問題点

KOYAMA Yu
東京都市大学環境情報学部情報メディア学科 2011 年度卒業生
SUWA Keisuke
東京都市大学環境情報学部教授

さらに、机上面で作業するために必要な明瞭等や光の反射，人工照明の角度，天空輝度の高さといった様々なものを考慮し，人工照明の配置設計とシステムの設計をする必要がある。

上記のような理由より，使用場所に応じた照明を制御するシステムが省エネルギー化および作業環境の改善に必要である。

1. 2 研究の目的

既存の「Arduino を用いた照明制御システム」^[2]では，大きな部屋や人がいない場所の照明点灯や消灯の忘れをなくすことで，消費電力を削減させることを目的としている。つまり，場所に対する無駄な照明を少なくするシステムである。さらに，照明の電源の ON/OFF ができるものを備え付けのスイッチや1つの照明に対して1つのリモコンで行うことの不便さによる照明の消し忘れの問題に対して，iPhone を利用することで，使いやすく，複数の照明をコントロールすることが有効である。これにより，場所に対する無駄な照明の消費電力削減と，リモコンの削減によるコスト減少ができる。しかし既存のシステムでは，iPhone に搭載されている機能があまり利用されておらず，マイコンとスマートフォンの組み合わせが活かしていないという課題がある。

本研究では，既存のシステムからマイコンとスマートフォンの組み合わせの新たな可能性を考え，iPhone, Arduino の機能をさらに引き出すための改良を行い，さらなる電力消費削減を実現させることを目的としている。電力消費削減の方法として，Arduino を利用した，LED の ON/OFF 点灯から，LED の輝度制御を行えるようにプログラムに変更を加える。さらに，iPhone の機能の利用として，内蔵された時計の利用によって，昼間の照明の明るさを制限する。また加速度センサーを利用することで，LED の段階的操作をより簡単に行う。これらの機能をシステムの構築により実証する。

2 提案する照明制御システム

2. 1 新しい照明システムの提案

照明制御システムの必要性や昼間の人工照明の必要性および問題点を考慮して，人工照明の配置およびシステム設計を行わなくてはならない。具体的には昼光とのバランスを考えた人工照明の輝度を自動調節し，さらに個人での制御も簡単にできるシステムがよいと考えられる。しかし，現在の照明制御システムは，スマートフォンで制御できるものがあるが，輝度を調節できない，輝度はシステムで制御されているが，個人で照明を変えることが難しいといった一長一短な

システムが多い。そこで，上記を考慮したより利便性の高いシステムの提案を行う。

図2は今回新しく提案する照明システムである。照明の操作は個人で操作できることと，コスト削減を考え，スマートフォンを利用する。部屋の照明配置レイアウトを画面上に表示し，そこに表示された各照明を加速度センサーを利用して連続調光方式で制御する。さらに窓際に配置した照度センサーを利用して，人工照明を制御する。さらに個人の机に対しても画面上からタッチすることで，机上の照明制御も可能とする。このように昼光を考慮し，さらに個人で簡単に自分が使う場所の照明を制御するシステムの構築ができるようにする。

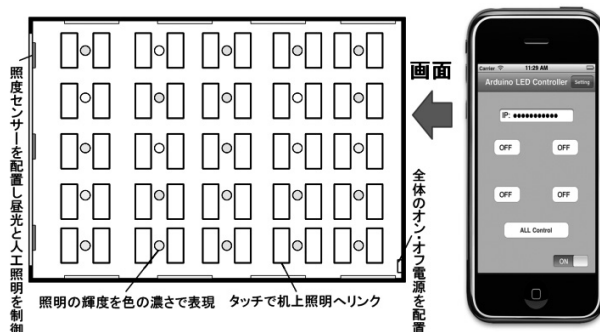


図2 新しい照明システム

2. 2 本研究のシステム

本研究では，提案したシステムの一部を Arduino と iPhone という組み合わせで作成し，照明実験システムを利用してシステムの評価と考察を行う。Arduino とは，8bit の AVR マイコンを利用したマイコンボードと，ArduinoIDE(統合開発環境)ソフトウェアからなるシステム開発環境のことである。スタンドアロンで動作し，シールドというモジュールを追加することで拡張が可能であり，自由度・拡張性に優れているという特徴がある。従来システムでは，1つの部屋の，個々の照明制御はできている。今回は，加速度センサーを利用した照明の輝度制御と，iPhone に搭載された機能等を引き出すことにより，操作性の向上とさらなる省エネを実現する。

3 照明制御方法

3.1 ネットワーク構成

本研究では、既存システムネットワーク構成を利用する。既存システムは iPhone と Arduino の間を WiFi で接続しており、本研究でも WiFi を接続回線として利用した。図3が既存のシステムで利用しているネットワーク構成である。iPhone を照明制御用のリモコンとして使用し、インターネットと Ethernet Shield を介することで、Arduino に LED を制御するためのパラメータを送っている。

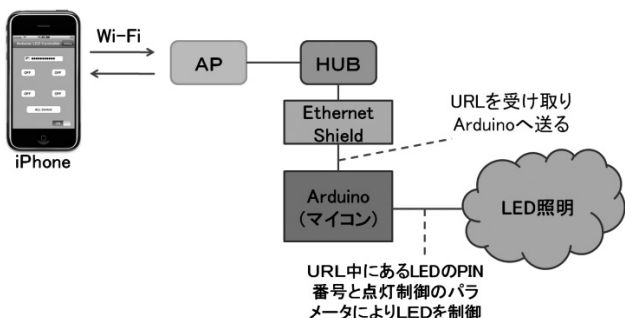


図3 ネットワーク構成

3.2 従来の照明システムとの比較

図4はシステム全体の流れを比較したものである。従来システムでは、iPhone での動作は、ON/OFF を URL に組み込み送信するのみであったが、本研究のシステムは、iPhone に搭載されている加速度センサーや時計を利用した制御を行っている。表1はその制御方法や機能、LED の個数の比較をわかりやすくまとめた表である。照明制御方法と輝度制御については従来シ

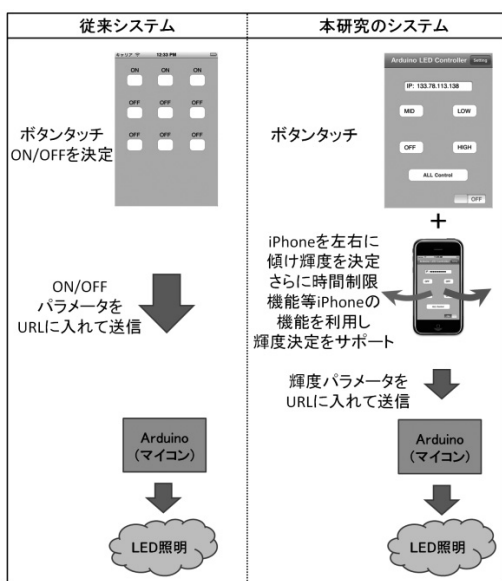


図4 従来のシステムと本研究のシステムの比較

テムではボタンタッチで ON/OFF 制御のみであったが、本研究のシステムはボタンタッチと加速度センサーを利用することによって、4段階制御かつより直感的な操作を実現している。また、本研究のシステムでは、新たに時間制限機能、一括制御機能を設けていて、それぞれ省エネ、操作性の向上を実現している。LED の個数に関しては PWM(パルス幅変調)を使用する関係上従来システムの9個から4個に減っている。

表1 システムの比較一覧

	従来システム	本研究のシステム
制御方法	ボタンタッチ	ボタンタッチ+加速度センサー
LEDの輝度制御	ON/OFFのみ	OFFを含めた4段階
時間制限機能	なし	あり
一括制御機能	なし	あり
LEDの個数	9	4

3.3 Arduino側のシステム

図5はArduino側での処理をフローチャートに表わしたものである。従来システムとの変更点は、輝度パラメータの確認部分である。従来は点灯・消灯のみだった部分を、点灯時の輝度を変化させることが可能となった。全体の流れにあまり変更がないようにプログラミングし、既存のシステムを活かしている。

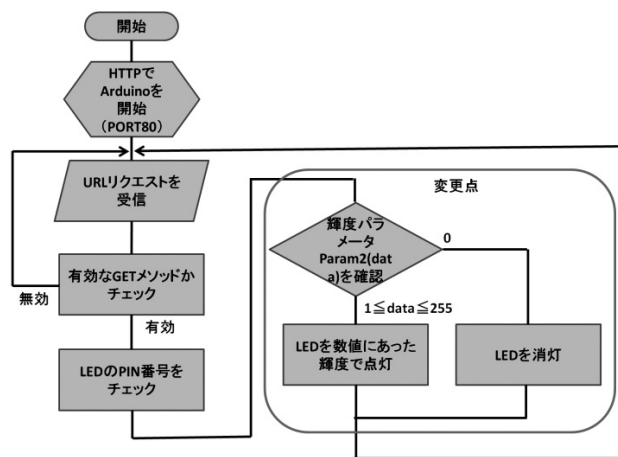


図5 Arduinoのフローチャート

3.4 PWM(パルス幅変調)

LED の制御には、PWM(パルス幅変調)を利用する。PWMとは、パルス信号を出力しておく時間(パルス幅)を変化させ、電圧・電流を制御する方法のことである。パルス信号とは ON/OFF を繰り返す電気信号のことで、ON にしている時間比率をデューティ比と呼ぶ。図

6のPWMによる制御に示すように、デューティー比を任意の値に設定して高速でON/OFF制御を行う。

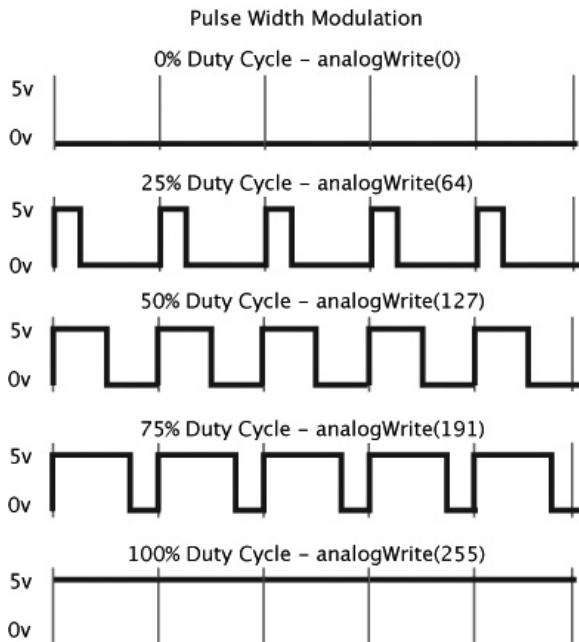


図6 PWMによる制御^[3]

PWMを利用すると、高速でLEDのON/OFF制御を行うので、デューティー比の値を小さく設定すると人間の目には暗くみえるようになる。本研究で使用しているArduinoではPWMを使用できるピンが限られており、3, 5, 6, 9, 10, 11番ピンのみとなっている。

ではなぜこのPWMを利用してLEDを制御するかを説明する。まず1つ目はLEDは従来のフィラメント式の電球と違い、光る電圧/電流範囲が狭いため、電圧/電流制御で明るさをコントロールするのが難しいためである^[4]。2つ目は、PWMは電力効率がよいからである。電圧/電流に負荷を与え制御を行う場合、無駄な電力が熱として発生してしまうが、PWMではON/OFFで制御するため電力に無駄がなく、理論上は電力効率が100%である^[5]。

3.5 iPhone側のシステム

図7に示すのがクライアント側アプリケーションとして今回新しく作成したiPhoneアプリケーションの画面と制御方法である。アプリケーションを起動すると左の画面になり、4つのLEDすべてがOFFの状態のLED個々制御用ボタンと一括制御ボタン、ON状態の時間制限スイッチ、IP欄が未設定で黒丸となっている状態で画面が表示される。

LED個々制御用ボタンは、そのボタンを押した状態で右図のようにある一定角度に傾けることで、OFF、LOW、MID、HIGHの4段階に画面のボタン表示とLED

の輝度を変化させることができる。これは加速度センサーを利用しており、一括制御ボタンは押した回数によって、4つすべてのLEDの輝度を制御することができる。時間制限スイッチはON状態ならば、設定時間の間の輝度を制限することができるスイッチとなっている。それぞれ輝度を制御する際は、URLをプログラム内で作成し、Arduinoに送信している。

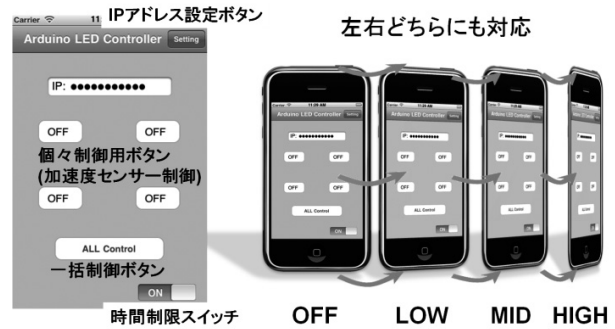


図7 iPhoneアプリ画面と制御方法

3.6 システム全体の流れ

システム全体のシーケンスを図8に示す。照明制御を行うユーザが、iPhoneの個々制御用ボタンをタッチすることで加速度センサーが起動し、iPhoneを任意の位置に傾げることで、iPhone内で輝度を段階的に変化させ、そこで決定された輝度でURLを作成し、Arduinoに送信する。Arduinoは送信されてきたURLに従いLEDの輝度をPWMで制御し、LEDを点灯させる。一括制御や時間制限機能がONとなっている場合などでも、iPhone内で全て処理し、Arduinoは点滅制御以外の制御は行っていない。このようなシステム全体の流れになっているため、システム変更があったとしても、ほぼArduinoでの変更はない状態となっている。

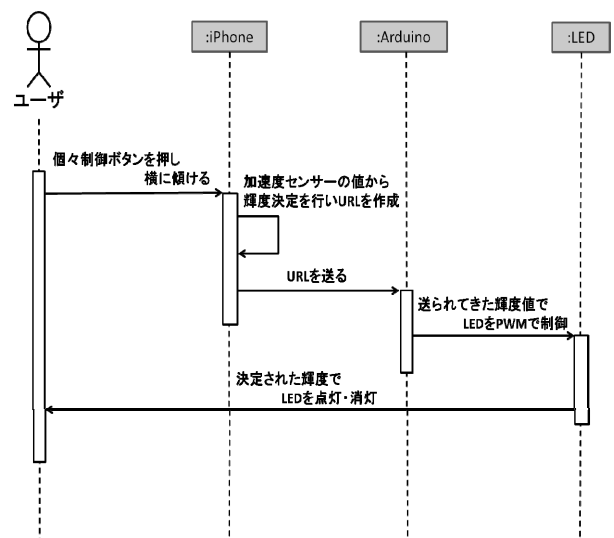


図8 システム全体のシーケンス

4 照明制御実験システムの構築と評価

4.1 照明制御実験システム

今回、研究を行う上で図9に示すように実際に照明制御実験システムを使用した。この実験システムは既存システムを利用したもので、PWMを使用して4個のLEDを制御することとした。

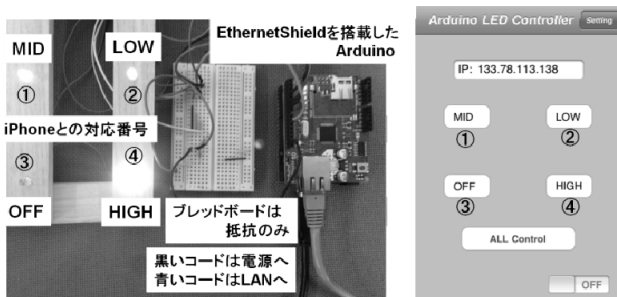


図9 照明制御実験システム

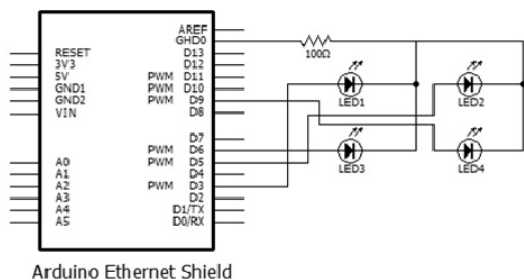


図10 照明制御実験システムの回路図

図9の写真では、OFFを含めた4段階の輝度制御を個々に行っている状態である。それに対応し、右のiPhoneアプリ画面の表示も切り替わっている。LEDの輝度制御はArduino側で送られてくるURLを処理してLEDを制御するというシステムから連続調光ではArduino側で処理できない可能性があったため段階的調光とした。LOW、MIDの調光の値は、それぞれLEDの明るさの変化が明確に現れる部分を何度も設定を変えることで決定した。左の写真の一番右側にあるのがEthernet Shieldを搭載したArduinoで、Arduinoの上にEthernet Shieldが載っている状態である。Ethernet ShieldはArduinoの拡張ボードの1つでネットワークに接続できるように拡張できる。PCなどを利用せずスタンドアロンでデータの送受信を単体で行うことができるが、デジタルI/Oピンの10, 11, 12, 13番を使用するので、使用できるデジタルI/Oピンが少なくなってしまう。真ん中にあるブレッドボードは抵抗のみで、ArduinoとLEDの中間の接続を行

っている。電源はArduinoの基板についているUSB-ACを利用し供給することができる。これによりPCを利用せずにスタンドアロンで動作することができる。本研究では、既存システムとの動作比較・評価を行ったので、この実験システムの回路は従来システムから引き継ぎ、簡素である。図10は既存システムから変更を加えた後のArduinoのEthernet Shield周辺の回路図である。LEDの輝度を制御するために、PWMを利用できるピンしか接続していないため、LEDの個数も4つとなっている。PWMを使用できるピンとして10, 11番ピンも残っているように見えるが、Ethernet Shieldを使用しているため、LEDの制御には利用できない。

4.2 制御プログラム

照明制御のArduino側のプログラムをArduino上でのデータのやり取りを示したシリアルモニターを利用して説明を行う。図11で行っている動作は9番ピンに繋がっているLEDを255(最大出力)で点灯させるという動作を実際に行った状態である。まず全体の流れを①～③に分けて解説する。

- ①URL中のどの部分を取得するかをあらかじめ指定してあり、その指定された部分を取得した部分である。
- ②取得した情報をスラッシュごとに分け、分けたデータをもとに、それぞれのパラメータの変数に割り振っている。
- ③割り振られた変数がcmd, param1, param2である。cmdはピンの出力方法を指定し、param1はピン番号、param2は出力の大きさである。

PWMを使用するため出力方式はanalogWriteを使用する。PWMをすべてに使用する場合はこのanalogWriteのみになるため、この出力方式の選択は必要ないが、今後digitalWriteを使用する可能性を考え、システムの中に残した。

また、最大出力255までの数値を扱うため、param2のデータを数値に変更する。URL形式で送られた情報を取得する際のデータ型はchar型(文字列)で、char型の数値認識の範囲は-128~127のため、最大数値を入れても正常に動作しない。そのためchar型からint型にデータを変更するため、atoi関数を利用した。

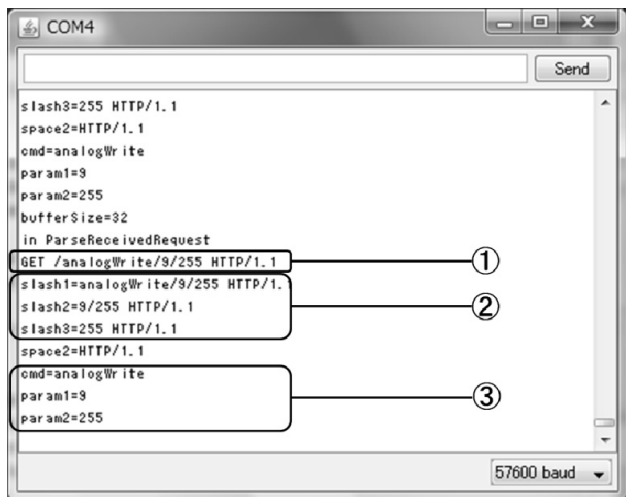


図 11 Arduino のシリアルモニター

iPhone 側プログラムは、個々制御ボタン、一括制御ボタン、時間制限機能でプログラムが分かれている。個々制御ボタンは加速度センサーを利用している。

各ボタンにタッチアクションに対する応答が3つ用意されている。ボタンを押されている状態であれば、加速度センサーが起動する。加速度センサーが起動すると x 軸, y 軸, z 軸の3方向から値が-1以上1以下でそれぞれ返される。今回は横に傾けた時に値が返る x 軸のみを利用する。x 軸から返される値は acceleration. x に格納される。その値から輝度の段階を決定し、決定した輝度段階数値を iPhone アプリ上の画面の文字と URL への組み込みから送信を行うメソッドに送り、Arduino に送信することで、LED を制御している。さらに時間制限機能を ON としている場合は、-1以上0.75未満, 0.75より上1以下の場合で分岐処理が行われ、最大出力にならないように設定される。

残り2つの動作は、ボタンが離された場合、何かしらによってキャンセルされた場合である。どちらの動作も加速度センサーを停止させる。ボタンが離された場合というのは、照明のコントロールを終えたときである。キャンセル動作は、たとえば着信などによって意図せずアプリ画面から切り替わってしまった場合などで、加速度センサーを停止しないと、エラー等を起こしてしまうためである。

加速度センサーの値により決定された輝度段階の数値0~3の値を受け取り、その値をもとに輝度を設定する。そして設定された輝度、ボタンをタッチしたときに得たピン番号、IP アドレス設定画面で入力された IP アドレスをそれぞれ順に連結させていき、URL を作成する。作成される URL は図 12 のようになる。

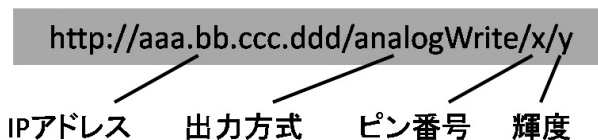


図 12 URL 作成例

時間制限機能はスイッチを利用して実装した。スイッチが ON の場合に、時間を取得し、取得された時間が制限範囲内であるかどうかを返すというプログラムである。時間取得する際に、setLocale を指定することで iPhone の時計表示設定が 24 時間表示・12 時間表示であろうとも正常に動くように設定してある。

スイッチが ON である場合、tm という変数が 1 に設定され、LED が HIGH で点灯するさまざまな場合において if 分岐として挿入され、そこで時間判定が行われるようになっている。

一括制御ボタンは押した回数を変数に格納し、その回数によって輝度の段階を変える仕組みとなっている。一括制御ボタンのみ、メソッドを独立して作成した。理由として、既存システム時より問題点が多く挙がっていたため、バグやエラーが独自に起こることが予測されたためである。一括制御ボタンメソッドでは、輝度設定と文字変更を行い、その情報をもとに一括 URL 作成送信メソッドで URL 作成と送信をすべての LED で行うようにした。

4. 3 従来システムと提案したシステムとの比較

従来の照明制御実験システムと本研究の照明制御実験システムの動作比較を図 13 に示す。

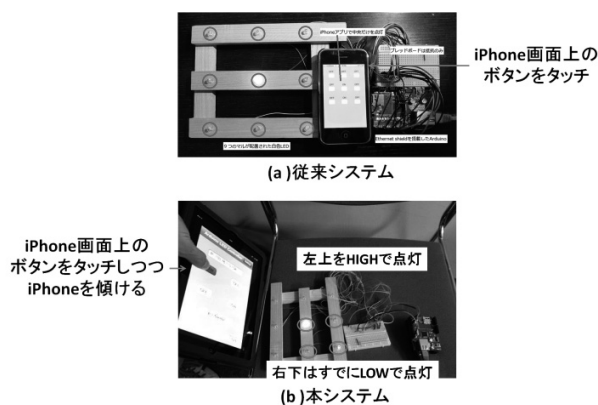


図 13 従来の照明制御実験システムとの比較^[2]

本研究の照明制御実験システムは従来の照明制御実験システムのシステム全体の流れを踏襲しているため、形状は同じものとなっている。しかし、LED が扱える個数、LED の輝度を制御・制限する方法等が大

大きく異なる。制御可能なLEDの個数は9個から4個に減少しており、本研究の照明実験システムである図13(a)写真には、9個のLEDが設置されているが図13(b)の丸がついている4個しか制御できない。LEDの輝度制御方法は図のように、従来システムでは、iPhone画面上のボタンをタッチするのみで、本研究のシステムではiPhoneを傾けることでLEDの輝度を段階的に制御できる。

本研究で作成したシステムは、1つの部屋の天井照明を制御する部分に該当する。実験システムの結果より個々の照明の輝度制御、時間制限機能といった機能を実装することができた。しかし、本研究のシステムでは、昼光を利用した照明制御が難しい。理由としては、照度センサーを搭載し、さらにLEDの連続調光可能なシステムを構築できなかった点にある。理由の詳細は4.4で述べる。提案したシステムにおいて、それぞれの部屋とのリンクや、机上照明とのリンクであるが、本研究でのシステムであれば、IPアドレスの設定を行うため設定するIPアドレスを変えるだけで他の部屋(LED)の制御を簡単に行うことができる。さらに画面遷移を行う場合も、iPhoneアプリケーションのプログラムの変更・追加をするだけで可能なため、実装が容易である。これはArduinoがiPhoneアプリケーションで作成した輝度等の情報をLEDに送るという役割のみであるというシステムの構造上、照明の追加を行う場合、Arduino側システムを複製しIPアドレスの新たな割り当てを行うだけで良いからである。また、Arduinoの単価はそこまで高くないため、コストは安く済む。

実験システムを動作させた結果から、提案したシステムのように、昼光を利用したシステムではないものの、個々のニーズにあわせ輝度制御が可能かつシステム全体としての柔軟性があるため、実際の天井照明を制御するシステムとして、十分使えるシステムといえる。

4.4 考察

本研究で作成した照明システムを照明制御実験システムで点灯させた結果、すべての機能の実装を確認できた。照明用のリモコンとして、加速度センサーを利用した直感的な操作で4つのLED照明の4段階の輝度制御を可能とし、さらに一括制御、時間制限機能といった操作性や省エネを意識した十分なシステムを構築することができた。しかし、既存システムの2つの問題点が解決できなかったこと、既存システムより制御するLEDの個数の減少、昼間の照明制御の理想的な輝度を実現できなかったという問題点が挙げられる。以下に詳しく記述する。

既存システムで挙げられていた問題点の1つ目は、一括制御の際の誤動作の問題である。既存システムでは一括制御の機能は設けられていなかった。理由として、一度に大量のURLを送った場合、Arduino側で処理しきれないことを危惧したためである。今回は機能の実装を行いその検証を行ったところ、危惧されていたように、処理しきれない状態に陥ることがあることを確認した。一括制御ボタンを連打した場合、Arduino側で処理することができず、LEDがイルミネーションのように様々な状態に点灯するという誤動作が起きてしまった。しかし、押す回数が1回の場合や連続で押す場合も連打しなければ問題なく動作することが確認できた。今後この機能を実際に使えるものとするためには、iPhone側で連続で押すことへの制限をかけるか、Arduino側のプログラムを変更し処理できるようにする必要があることがわかった。

2つ目はArduinoとiPhoneアプリの同期問題である。個々の制御をする際はほぼiPhone画面と点灯しているLEDが対応していたが、一括制御ボタンの誤動作などによって、実際の点灯状態と同期しないという問題があった。しかし、この問題点に関しては、照明制御という観点において、実際に自分の目で確かめるため、Arduino側からの通知は必須ではないと考える。もちろん何らかの方法でArduino側からの応答があったほうが他システムへの利用などを考えた場合必要であると考えられる。

既存システムよりLEDの個数が減少してしまったという課題は、PWMが利用できるピンの個数の問題であるが、使用しているマイコンやプログラムに柔軟性をできるだけ持たせるように設計したため、Arduinoの種類や個数を変えることで簡単に解決できるのではないかと考える。

昼間の照明制御の理想的な輝度を実現できなかったという問題点は、iPhoneに搭載されている環境光センサーがアプリケーションとして利用できないということに起因する。昼間の最適な照明は1.1で述べたとおり、昼光の利用や、照明の向きを考慮した輝度を設定しなくてはならない。これを行うためには、昼光の輝度や暗くなってしまう部分の輝度を感知し、その値からLEDの輝度を計算しなくてはならない。その感知方法がiPhoneでは利用できなかったため、実装するにいたらなかった。

5 おわりに

5.1 まとめ

本研究では、既存システムの改良を行い、操作性の向上とさらなる省エネルギーの実現を目的として研究を行った。実際に Arduino で LED の輝度制御を行うためのプログラムの変更を行い、新しく iPhone のアプリケーションの作成を行うことで、LED の点滅および輝度の制御を可能とした。これにより昼間等明るさが最大である必要がない場合の輝度調整が可能となり、さらに内蔵時計の機能を ON にすることで意識しなくても明るさを制限でき、その分省エネとなりさらに操作性も向上することができた。

5.2 今後の課題

今後の展望としてまず挙げられるのは、昼間における理想の照明システムの実現である。iPhone と Arduino の組み合わせによる効果は最大限に発揮できてないと考えており、iPhone で使えない環境光センサーなどを Arduino 側で使用することで、解決することができるのではないかと考えている。また iPhone アプリ側で照明に対する細かい設定を行い、Arduino を利用した照明システムを部屋ごとに配置することで、1つのアプリケーションで容易に様々な部屋の照明を制御することが可能である。

また、このシステムでは iPhone と Arduino という特定のスマートフォンとマイコンでの組み合わせで行ったが、他のスマートフォン、特に普及が爆発的な勢いで進んでいる Android での利用環境を整えていくことで、コスト面などから有効であると考えられる。

さらに、本研究によってマイコンとスマートフォンの両方を活かすことで、省エネルギー化や操作性の向上が可能であるということが証明できた。今後、ホームオートメーションなども、マイコンとスマートフォンで構築することで、省エネルギー化、生活の快適化を行うことが可能である。

謝辞

本研究を進めるにあたり、システム作成の際に協力して頂いた諏訪研究室 OB の柴田吉範氏、同研究室 4 年生の小野澤清人氏に感謝の意を表します。

参考文献

- [1] インプレス R&D,
<http://www.impressrd.jp/news/111108/kwp2012>
- [2] 柴田吉範, Arduino を用いた照明制御システム, 東京都市大学環境情報学部諏訪研究室平成 22 年度卒業論文 2010.3
- [3] Arduino チュートリアル 基礎編,
<http://www.musashinodenpa.com/arduino/ref/index.php?f=2&pos=152>
- [4] PIC で LED の調光,
<http://www5b.biglobe.ne.jp/~YASUSI/gallery/electronics/050828/050828.htm>
- [5] 片山泰男, 1bit アンプ, 原理と作成,
<http://home.catv.ne.jp/dd/pub/pwm.html#>