

スマートフォンのフラッシュライトを用いた 可視光通信によるIoTデバイス設定手法の検討

濱本 時空¹ 乃村 能成²

概要：IoTデバイスの導入時には初期設定が必要で、デバイスの数に比例してその手間が増大する。しかし、初期設定の手間を削減するためにデバイスを高機能化すると、価格が上昇してしまい、IoTデバイス導入の障害となりうる。そこで、IoTデバイスの設定においてLEDを用いた可視光通信を利用することで、価格上昇を抑えつつ初期設定の手間を削減する。この可視光通信による初期設定手法を、すでに所持しているスマートフォンで可能になれば、IoTデバイス導入をより簡単なものにできると考える。しかし、スマートフォンのフラッシュライトは可視光通信用として設計されていないため、実際に可視光通信に使用できるかは分かっていない。そこで、スマートフォンのフラッシュライトの特性について調査し、可視光通信への有用性を調査する。また、スマートフォンのフラッシュライトによる可視光通信手法を用いて、IoTデバイスの設定を行うシステムの設計および実装を行う。

キーワード：IoT, 可視光通信, マイクロコントローラ, スマートフォン

1. はじめに

IoT (Internet of Things) デバイス本体の価格と初期設定手法の良し悪しは、導入のしやすさに影響する。したがって、初期設定を行いやすくしつつ、IoTデバイスの価格上昇を抑える必要がある。そこで、スマートフォンを用いたIoTデバイス設定手法について考える。すでに所持しているスマートフォンでIoTデバイスの初期設定が可能になれば、IoTデバイス導入時に必要な機材を持ち運ぶ必要がなくなるため、IoTデバイス導入をより簡単なものにできる。スマートフォンでIoTデバイスの設定を行う方法として、Google Home[1]のシステムを用いた設定やBluetoothによる設定等が挙げられる。しかし、これらの設定方法は安価で低機能なIoTデバイス向きではない。Google Homeによる設定方法は特定のIoTデバイスしか設定できない。Bluetoothによる設定方法は、IoTデバイスにBluetooth通信に必要なハードウェアが必要となり、IoTデバイスの価格が高くなる。これらのことから、低価格低機能のIoTデバイスをスマートフォンで設定する汎用的な手法が必要だと考える。

先行研究としてIoTデバイスに接続したLEDを用いた可視光通信を行う手法[2]がある。既存手法では、IoTデ

バイスに接続したLEDを可視光通信の送信部と受信部として用いる。LEDは、順方向に電流を流すと発光するという性質を持つが、逆にLEDに光を当てると電圧を生じるといった性質を持つ[3]。この性質と、マイクロコントローラに備わっている入力と出力を切り替えられる入出力端子を用いることで、1つのLEDを用いて可視光通信が可能になる。既存手法の送信部のLEDをスマートフォンのフラッシュライトで代替できれば、IoTデバイスの価格増大を抑えつつ初期設定の手間を削減できると考える。

しかし、スマートフォンのフラッシュライトは可視光通信用として設計されていないため、実際に可視光通信に使用できるかは分かっていない。そこで、本研究ではスマートフォンのフラッシュライトについて調査し、スマートフォンのフラッシュライトが可視光通信においてどの程度の有用性があるかを調査する。

2. スマートフォンを用いた可視光通信によるデータ送信手法

2.1 スマートフォンのフラッシュライトによる可視光通信

スマートフォンを用いて可視光通信を行う方法として、IoTデバイスのLEDにスマートフォンのフラッシュを当てて設定する方法を考える。フラッシュライトで可視光通信を行う場合、以下の2点の問題がある。

問題1 正確な点滅時間の制御

¹ 岡山大学大学院自然科学研究科

² 岡山大学学術研究院自然科学学域

可視光通信において、ビットレートを向上させるためには高速で点滅を切り替えなければならない。例として、既存手法の可視光通信プロトコルでは、10文字分のデータを1秒未満で送信するために5ms単位で点滅の時間制御が必要となる。フラッシュライトの点滅周期が遅い場合、初期設定にかかる時間が増加してしまう。また、点滅周期が短くなるにつれ、点滅を切り替える精度も重要となる。ソフトウェア上での点灯から消灯に切り替える処理が遅れてしまうと、設定した時間よりも長い光信号を送ってしまうことになる。AndroidをはじめとするスマートフォンはリアルタイムOSではないため、時間の計測の粒度が粗く、数十ミリ秒単位で時間を計測している場合がある [4]。また、Androidでは処理が数ミリ秒単位で遅れることが報告されている [5]。したがって、ソフトウェア上で設定している点滅周期よりも、実際のフラッシュライトの点滅周期の長さが異なる可能性がある。点灯と消灯の長さで0と1のデータを判別するプロトコルの場合、この長さの違いによって誤ったデータを送信することになってしまい、通信精度に大きな影響を及ぼす。

問題2 フラッシュライトの光の強さ

可視光通信を行うためには、点灯と消灯の判別ができる明るさの光が必要となる。受信側とスマートフォンの距離および位置によって、受光する光の強さは大きく変化する。スマートフォンをIoTデバイスにかざして設定を行う場合、スマートフォンとIoTデバイスとの距離はフラッシュライトの強さに依存する。しかし、許容されるLEDへの光の入射角は狭く、受光LEDとスマートフォンのフラッシュライトの光軸がずれると、LEDに入る光の量が大きく減少してしまう。また、スマートフォンのフラッシュライトの位置は機種によって大きく異なり、受光LEDとの光軸を正確に合わせるのは困難であるため、実際のフラッシュライトの光の強さに対して、LEDに届く光の強さは低くなると考える。さらに、スマートフォンとIoTデバイスとの距離によっては、周囲の環境光等がLEDに入りやすくなり、周囲の明るさが可視光通信の精度に与える影響が大きくなると考える。このため、フラッシュライトの光の強さが強くないと光信号の判別が難しいと考える。

以上の問題点を解決するための検討を以下に示す。

検討1 可視光通信プロトコルの改良

フラッシュライトを点滅させて光信号を送信するアプリケーションを作成し、それをIoTデバイスのLEDで受光した際のデータを数値化することで、フラッシュライトがどの程度高速に点滅できているかを調査する。また、スマートフォンとLEDの距離を変えな

がらIoTデバイスで光信号を受光し、その際の光の強さを数値化することで、フラッシュライトの光の強さを調査する。調査で得られた結果から、スマートフォンのフラッシュライト向けに既存手法の可視光通信プロトコルを改良する。

検討2 受信側の光の強さ読み取り精度向上

スマートフォンの距離および位置が変化することLEDに入る光の強さが弱くなる。この問題に対処するために、受信側での光の強さの読み取り精度を向上することで、弱い光でも正しく光信号を判別できるようにする。

2.2 スマートフォンのフラッシュライトの特性調査

2.2.1 概要

一定周期でフラッシュライトの点滅を繰り返すAndroidアプリケーションを作成し、そのフラッシュライトの光をIoTデバイスで受光する。受光時の光の強さを数値化することでフラッシュライトの点滅制御の精度および光の強さを調査する。

2.2.2 調査方法

スマートフォンのフラッシュライトをどの程度正確に制御できるかを調べるために、スマートフォンを送信側、IoTデバイスを受信側として、フラッシュライトの特性の調査を行う。送信側および受信側の詳細を以下に示す。

(送信側) スマートフォン

フラッシュライトを点滅させるAndroidアプリケーションを作成し、フラッシュライトを一定周期で点滅させる。今回使用したアプリケーションでは、フラッシュライトの点滅周期を1ms単位で設定可能になっている。点滅周期は、たとえば1msに設定すると点灯1msと消灯1msを繰り返す。

(受信側) ESP32

IoTデバイスであるESP32にLEDを取り付け、光をLEDで受光して生じた電圧を値として読み取る。1章で述べたように、LEDは光を受け取ると電圧が発生する性質を持つ。よって、LEDから発生した電圧の高低で、光が当たっているかどうかを判別できる。LEDで発生した電圧の読み取りは、ESP32に内蔵されているA/Dコンバータで行う。ESP32に内蔵されているA/Dコンバータは分解能が12bitで、0~0.95Vまでの電圧を、0~4095の数値として読み取れる。読み取り周期は、スマートフォン側で設定できる最短点滅周期である1msよりも短い、0.2ms間隔とする。

2.2.3 調査手順

フラッシュライトの特性調査の手順を以下に示す。

- (1) スマートフォンを受信側の受光LEDに一定の距離を離してかざす。

表 1 スマートフォンアプリケーションの実装環境

通番	スマートフォン機器名	OS
1	Nexus 5X	Android 6
2	Galaxy S9	Android 10
3	Xperia XZ2	Android 10
4	Pixel 3a	Android 11

- (2) スマートフォン側で Android アプリケーションを操作し、フラッシュライトを一定間隔で点滅させる。
 - (3) ESP32 側で光の強さを読み取るプログラムを実行し、IoT デバイスに取りつけた LED でスマートフォンのフラッシュライトの光を受光する。
 - (4) LED で発生した電圧値を 0.2 ms 間隔で 20,000 回計測し、フラッシュライトの光の強さの変化を計測する。
- 以上の計測を行うことで、ソフトウェア上で設定している点滅周期と、実際のフラッシュライトの点滅周期との間にズレが生じているかを調べられる。そして、調査結果からフラッシュライト制御の精度を知ることができる。計測の際に変化させる条件を以下に示す。

- (1) 送信側は、点滅周期を 1ms から 10ms の間で、1ms 単位で変化させて計測を行う。今回実装した方法では、ソフトウェア上で設定できるフラッシュライトの最短点滅周期が 1ms であったため、変化させる点滅周期の範囲を 1ms からとしている。
- (2) 送信側と受信側の距離を、それぞれ 5cm, 10cm, 20cm の距離で変化させて計測を行う。距離の決め方は、ユーザが無理なく近づけられる距離から、複数台同時に受光できる距離までとした。

ここで、スマートフォンの機種によってフラッシュライトの点滅制御の精度や光の強さに差があると考えられる。このため、異なる機種間でのフラッシュライトの特性の差についても調査する。計測に使用したスマートフォンと OS のバージョンを表 1 にまとめる。今回は 4 機種を用いて計測を行った。

2.3 フラッシュライトの特性調査結果

2.3.1 電圧値のデータ

2.2 節で述べた計測方法で計測した電圧値のデータを時間軸に沿って並べたものの一部を図 1 に示す。図 1 は、Nexus 5X の機種において、受信側との距離が 5cm、ソフトウェア上の点滅周期が 1ms の条件で、フラッシュライトの光を LED で受光した際の電圧値の変化を表したグラフとなっている。縦軸が LED の電圧を A/D コンバータで読み取った値 (以降これを単に電圧値と呼ぶ)、横軸が時間 (ms) となっている。計測された電圧値のピークはおおよそ 1600 で、電圧にすると約 0.4V となる。図 1 から、以下の 2 つのことが分かる。

- (1) スマートフォンのフラッシュライトが点灯している部

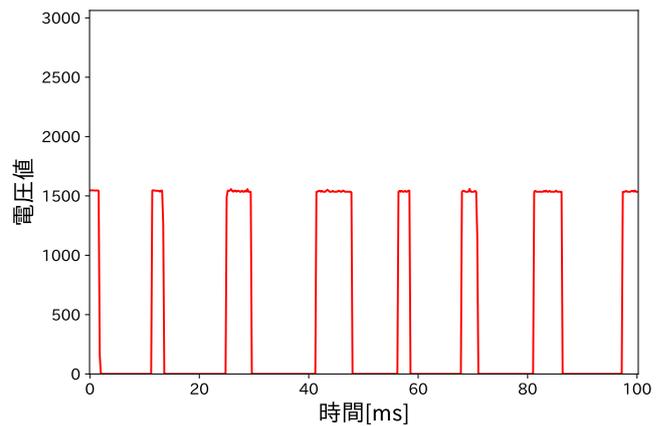


図 1 Nexus 5X における受信側との距離が 5cm、ソフトウェア上の点滅周期が 1ms の条件での電圧値の波形

分と消灯している部分を電圧値の大小によって見分けられる

- (2) グラフの形状を見ると、点灯時間と消灯時間が一定ではなく、さらに設定した点滅周期である 1ms よりも長く点灯と消灯している

フラッシュライトの点灯時間および消灯時間がソフトウェア上で設定した点滅周期とどれだけズレているかを見るために、今回計測された電圧値から点灯時間および消灯時間を抜き出し、ズレの統計を取る。図 1 の電圧値の波形を見ると、消灯時には電圧値は 0 の値を取っているため、電圧値が 0 の期間を消灯時間として抜き出し、それ以外の電圧値が 1 以上の期間を点灯時間として抜き出す。そして、それぞれ抜き出した点灯時間および消灯時間のヒストグラムを作成し、設定した点滅周期と実際のフラッシュライトの点滅周期とのブレ幅を確認する。

2.3.2 点滅周期のズレのヒストグラム

ソフトウェア上で設定した点滅周期と実際のフラッシュライトの点滅周期のズレをヒストグラムとしてまとめる。点灯時間および消灯時間のズレは、((点灯時間または消灯時間) - ソフトウェア上で設定した点滅周期) という式で求め、計測された点灯時間および消灯時間がソフトウェア上で設定した点滅周期とどれだけ異なるかを求める。各スマートフォンのフラッシュライトの点灯時間及び消灯時間のズレのヒストグラムを図 2 から 図 4 に示す。Galaxy S9 と Xperia XZ2 については、ズレの特徴が似ていたため、Galaxy S9 のヒストグラムのみを載せている。図 2 が Nexus 5X、図 3 が Galaxy S9、図 4 が Pixel 3a の点灯時間および消灯時間のズレのヒストグラムとなっている。各図において、縦軸が度数、横軸が時間 (ms)、何も模様がついていない部分が点灯時間のズレのヒストグラム、斜線で模様付している部分が消灯時間のズレのヒストグラムとなっている。グラフ中の破線が点灯時間のズレの平均値、点線が消灯時間のズレの平均値を示している。左から右に向かって順に、点滅周期を 1ms, 5ms, 10ms に設定した際

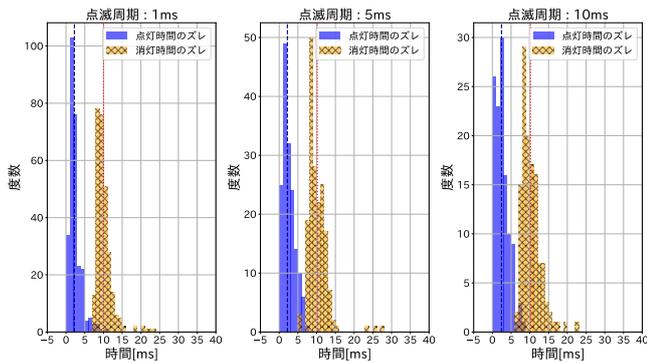


図 2 Nexus 5X における 1ms, 5ms および 10ms の点滅周期での点灯時間と消灯時間のズレのヒストグラム

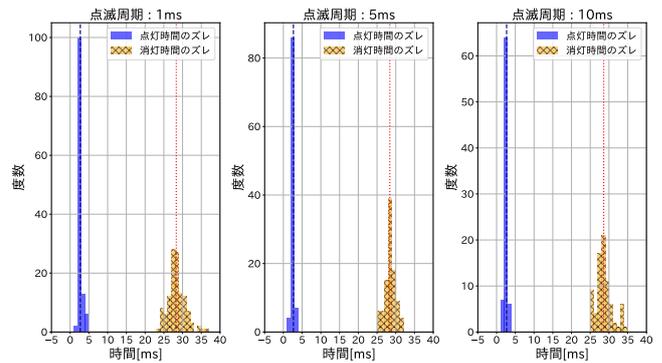


図 4 Pixel 3a における 1ms, 5ms および 10ms の点滅周期での点灯時間と消灯時間のズレのヒストグラム

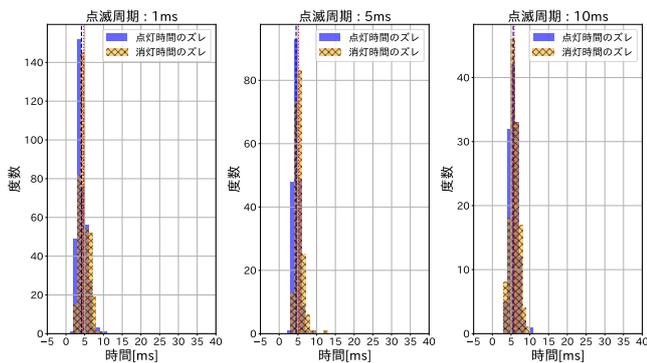


図 3 Galaxy S9 における 1ms, 5ms および 10ms の点滅周期での点灯時間と消灯時間のズレのヒストグラム

のヒストグラムとなっている。図 2 から 図 4 で分かることを以下に 3 つ示す。

- (1) スマートフォンの機種で点灯時間および消灯時間のズレの特性が異なる
- (2) Nexus 5X (図 2) と Pixel 3a (図 4) は点灯時間のズレよりも消灯時間のズレの方が大きい
具体的には、Nexus 5X において、点灯時間のズレの平均値は約 2.4ms で、消灯時間のズレの平均値は約 10ms となっていた。
- (3) Galaxy S9 (図 3) は、点灯時間および消灯時間のズレのヒストグラムが重なっていることから、点灯時間および消灯時間のズレのピーク、およびズレの振幅がほぼ同じである

これらのことから、点灯時間の方がズレの平均値が小さいため、点灯時間の長さでデータの 0 と 1 を判別するようなプロトコルが良いと考える。

各スマートフォンにおける点灯と消灯のズレの最大値、最小値および平均値をまとめたものを表 2 に示す。表 2 を見ると、今回計測に使用したスマートフォンの中で点灯のズレの最大値が一番大きい機種は Nexus 5X ということが分かる。このことから、一番ズレの大きいスマートフォンである Nexus 5X でも正しく可視光通信を行えるように、プロトコルの改良を行う。

2.3.3 フラッシュライトの光の強さと距離の関係

各スマートフォンにおける、フラッシュライトの光の強さと距離の関係を表 3 に示す。表 3 の数値は、フラッシュライト点灯時の電圧値の平均値を示している。表の中でハイフンになっている部分は、その測定条件では受信側に届く光が弱く、電圧値を測れなかったことを示している。表 3 から分かることを以下に示す。

- (1) どのスマートフォンでも距離が 5cm の場合、電圧値を計測できている
- (2) どのスマートフォンでも距離が離れていくにつれて電圧値が小さくなっている
- (3) スマートフォンの機種の違いによってフラッシュライトの光の強さは大きく異なり、全ての機種において距離を離して可視光通信を行うことは難しい

これらのことから、スマートフォンのフラッシュライトで LED を用いた可視光通信を行うことは可能であると言える。スマートフォンと受信側を近い距離に保つことは、正しくデータを送信するために重要となるため、全てのスマートフォンで IoT デバイスを複数台同時に可視光通信で初期設定することは困難であると言える。

2.4 受信側の光の強さ読み取り精度の向上

2.3.3 項で述べたように、スマートフォンのフラッシュライトの光を受信側の LED で受光する際に、スマートフォンの距離によって受光する光の強さが下がる。このため、受信側で光の強さの読み取り精度を向上させる必要がある。ESP32 で LED の電圧値を読み取る際に、読み取った電圧値が揺らいでしまい、正しく光信号を受信できないという問題があった。具体的には、フラッシュライトを消灯しているにもかかわらず、電圧値が 0 から 1600 ほどの間で揺らいでしまい、正しくフラッシュライトの点灯と消灯を判別できない場合があった。この問題を解決するために、LED の電圧値を読み取る前に、一瞬だけ ESP32 のポートの極性を出力に変更し、出力値を 0 とする。つまり、LED のアノードをグランドにつないで LED 内の電荷を 0 にするようにした。LED に溜まっている電荷を 0 にした後に

表 2 各スマートフォンにおける点灯時間と消灯時間のズレの最大値, 最小値および平均値

スマートフォン機種	ズレの最大値		ズレの最小値		ズレの平均値	
	点灯	消灯	点灯	消灯	点灯	消灯
Nexus 5X	25.8ms	47.0ms	-2.0ms	5.2ms	2.2ms	10.3ms
Galaxy S9	12.6ms	18.0ms	2.0ms	2.6ms	4.7ms	5.2ms
Xperia XZ2	8.0ms	10.4ms	0.6ms	2.0ms	3.7ms	4.6ms
Pixel 3a	11.0ms	39.4ms	1.2ms	22.4ms	2.5ms	28.1ms

表 3 各スマートフォンにおける距離の違いによる電圧値 (フラッシュライトの光の強さ) の変化

スマートフォン機種名	距離		
	5cm	10cm	20cm
Nexus 5X	1601	1123	188
Galaxy S9	1159	246	-
Xperia XZ2	107	-	-
Pixel 3a	1508	467	36

電圧値を読み取ることで, 電圧値の読み取りが安定化した.

3. スマートフォンを用いた可視光通信のプロトコル

3.1 プロトコルの改良

2.3 節で述べたように, スマートフォンのアプリケーションで 1ms 周期で点滅させようとしても, 点滅切り替えに時間がかかってしまい, 実際には点滅周期が数 ms 単位でズレていることが分かった. このことから, 既存手法の LED 可視光通信プロトコルでは, 正確に基準時間 T だけ光を点滅させられることが前提であったため, これをこのままスマートフォンのフラッシュライトで代替すると通信精度が悪くなると考えられる. そこで, 通信精度を向上させるため, スマートフォンのフラッシュライト用に通信プロトコルを改良する. 改良点を以下に示す.

(1) リーダの長さを $1T$ ではなく, 点灯 $8T$, 消灯 $4T$ にして, 基本時間 T を求めつつ, 明暗の閾値を求める. この改良により, 以下の 3 つの問題が解消される.

(a) 点滅制御にかかる時間の影響の緩和

既存手法のプロトコルでは, $1T$ のリーダーで信号の開始を伝えつつ, 基本時間 T を伝えていた. しかし, スマートフォンのフラッシュライトは点滅制御に時間がかかることがあるため, 正しく基本時間 T を伝えられず, 通信が失敗する恐れがある. このため, リーダの長さを $8T$ まで伸ばすことで, 点滅時間のズレが基準時間 T に及ぼす影響を小さくする.

(b) 外部からのノイズによる通信開始の誤検知防止

既存手法のプロトコルでは, 受信側は LED から閾値以上の電圧値を受け取ると, それをリーダーとして検知して光信号の受信を開始する. このため, 外部からの光をリーダーと誤検知してしまい, 予期

せぬタイミングで光信号の受信を開始してしまう場合があった. この問題を解消するため, リーダに $8T$ の点灯時間だけでなく, その半分の $4T$ の消灯時間も含めるようにする. こうすることで, 外部からの光を受信側が受け取ったとしても, それをリーダーだと誤検知する可能性を低減できる.

(c) 明暗の閾値の自動設定による通信の安定化

既存手法のプロトコルでは, 受信側で光の点滅を判別するための閾値は固定値だった. このため, 周囲の明るさの変化に対応できず, 通信精度が安定しないという問題があった. この問題を解消するため, リーダに点灯と消灯の 2 つの状態を持たせる. これにより, スマートフォンのフラッシュライトの光を判別する閾値を自動的に設定できる. また, スマートフォンを受光 LED にかざしたときの明るさが, 実際に可視光通信を行う際の消灯時の明るさである. したがって, かざした状態での消灯時の明るさと点灯時の明るさを閾値設定に用いることで, より正確な明暗判別ができる. 一つスマートフォンのフラッシュライトからの光信号を受け取ったか検知するためには, 受信側は起動時に一定時間 LED の電圧値を読み取って閾値のベースを設定し, その閾値から急激に変化した時点を光信号の開始地点とする.

(2) リーダ部分の消灯時間を測定できるようにデータ部分を変更する. (1) で述べたようにリーダーの消灯時間も基準時間に含める場合, 正しく消灯時間を計測できるようにデータの点滅パターンを変更する. 具体的には, 既存手法ではデータの点滅パターンは消灯後に点灯していたが, 点灯後に消灯するように変更する.

3.2 プロトコルの実装

3.1 節のプロトコル改良点を元に, スマートフォンのアプリケーションとしてプロトコルの実装を行う. 2.2 節で使用した Android アプリケーションと同じように, アプリケーション上で基準時間 T の長さを設定可能な可視光通信アプリケーションを実装した.

表 4 プロトコル評価環境

通番	スマートフォン機器名	OS
1	Nexus 5X	Android 6

3.3 プロトコルの評価

3.3.1 概要

3.1 節で行ったプロトコル改良によってどの程度通信精度が向上したかを評価する。スマートフォンから送信したデータと、受信側が光信号を受信して判別したデータとの正誤を算出し、データのビット単位での正解率を計測する。

3.3.2 計測手順

計測は以下の手順で行う。

- (1) スマートフォン側から特定の 10bit のデータを 100 回送信する。
- (2) 受信側は光信号を受け取り、データ部のデータを判別する。
- (3) (1) で送信したデータと (2) で判別したデータとの正誤をビット単位で測る。
- (4) (1) から (3) までを基準時間 T を 1ms から 10ms で 1ms 単位で変化させたものと、基準時間を 20ms, 30ms, 40ms に設定して計測する。

以上の計測手順を改良前のプロトコルと改良後のプロトコルで行い、ビット単位での正解率の観点からどの程度精度が向上したかを確認する。

表 4 に、改良したプロトコルの評価を行う環境を示す。使用するスマートフォンとして、2.2 節でフラッシュライトの特性を調査した際、一番点灯時間のズレが大きかった Nexus 5X を用いて評価を行う。

3.3.3 計測結果

計測結果を図 5 に示す。縦軸がデータのビット単位の正解率、横軸が基準時間 T を表している。図 5 において、破線が改良前のプロトコルを使用した場合の正解率で、実線が改良後のプロトコルの正解率を表している。図 5 を見ると、どの基準時間においても改良後のプロトコルの方が正解率が向上していることが分かる。精度向上の度合いとして、基準時間が 1ms のとき 25 ポイント、2ms のとき 31 ポイント向上しており、一番基準時間が長い 40ms のとき 3 ポイントの向上となった。特に、基準時間が短い部分ほど正解率が大きく向上している。この結果から、可視光通信プロトコルの改良は通信精度向上に効果があったと言える。

しかし、図 5 を見ると、改良後のプロトコルにおいて基準時間が 10ms 以降の正解率がほぼ一定になっている。表 2 の各スマートフォンにおける点灯時間のズレの時間を見ると、Nexus 5X の点灯時間のズレの平均値は 2.2ms であり、このズレの平均値よりも基準時間 T の長さを大きくしていけば、正解率は最終的に 100% になると考えられる。しかし、基準時間が 6ms 以降の精度向上率が低く、基

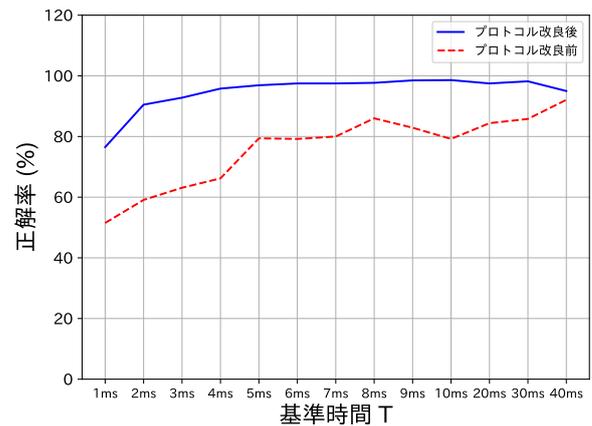


図 5 プロトコル改良前とプロトコル改良後の各基準時間 T における正解率

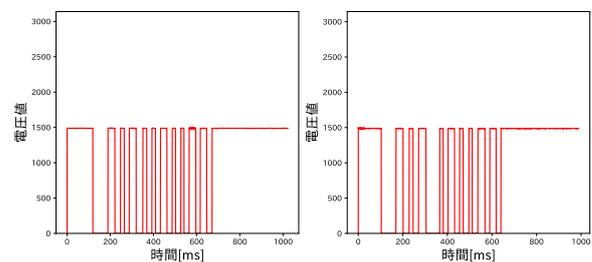


図 6 Nexus 5X におけるソフトウェア上の点滅周期が 10ms の条件での電圧値の波形の一部 (左: 正常な波形, 右: フラッシュライトが正常に点灯しなかった場合の波形)

準時間を 40ms にしても正解率が 100% にならないのは、プロトコルとは別の問題があるためだと考える。

図 6 に、Nexus 5X においてソフトウェア上の点滅周期が 10ms の条件で 10bit のデータを送信した際の電圧値の波形の一部を示す。左側が正常にフラッシュライトが点灯した場合の波形で、右側がフラッシュライトが正常に点灯しなかった場合の波形である。図 6 を見ると、一部分だけフラッシュライトが点灯していない箇所があり、この問題が正確さの評価に影響していた。フラッシュライトが点灯しない問題については、今後詳しい原因を特定する必要がある。

4. スマートフォンの可視光通信による IoT 初期設定手法

4.1 IoT デバイス初期設定の流れ

3 章でスマートフォンのフラッシュライトを用いて可視光通信を行うためのプロトコル改良を行った。このスマートフォンによる可視光通信手法を IoT デバイスへの初期設定に応用する。スマートフォンのフラッシュライトを用いた可視光通信によって IoT デバイスの初期設定を行う手順を図 7 を用いて説明する。

図 7 では、IoT デバイスの初期設定を行う例として、ユーザーがスマートフォンを用いて IoT デバイスに Wi-Fi の設定を行う手順を示している。設定手順を以下に説明する。

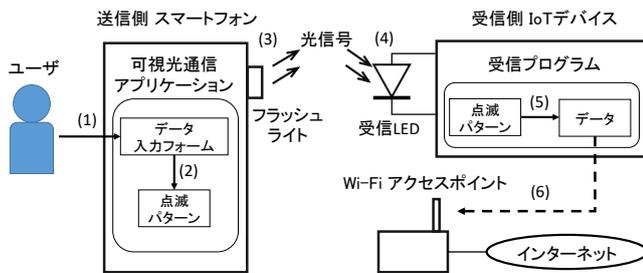


図 7 スマートフォンの可視光通信による IoT デバイスの Wi-Fi 設定時の手順

- (1) ユーザはスマートフォンの可視光通信アプリケーションに Wi-Fi の情報 (SSID およびパスワード) を入力する。
 - (2) スマートフォンの可視光通信アプリケーションが、可視光通信プロトコルに基づいて入力データを点滅パターンにエンコードする。
 - (3) 光信号の点滅パターンに基づいて、スマートフォンのフラッシュライトの点滅を制御する。
 - (4) 受信側の IoT デバイスの LED で光信号を受光し、光の点滅パターンを LED の電圧値として計測する。
 - (5) 計測した点滅パターンをデコードし、Wi-Fi の情報を取得する。
 - (6) 取得した Wi-Fi の情報を元に、Wi-Fi アクセスポイントに接続する。
- 以上の手順で、IoT デバイスの初期設定を行う。

4.2 IoT デバイス初期設定システムの設計

IoT デバイスに初期設定を行うためのシステムを設計する。IoT デバイス初期設定システムにおける、送信側と受信側の設計を以下に示す。

- (送信側) 送信側は、ユーザが入力した文字列を可視光通信プロトコルに基づいた点滅パターンに変換し、その点滅パターンをフラッシュライトの点滅により光信号として送信する。
- (受信側) 受信側は、スマートフォンからの光信号を受け取って、光信号のデータを元に初期設定を行う。LED の特性を活かし、LED を光信号の受信センサだけでなく、発光させてインタフェースとして用いることで、設定が完了したかを分かるようにする。

4.3 IoT デバイス初期設定システムの実装

4.2 節の設計に基づき、IoT デバイスの初期設定として Wi-Fi 設定を行うシステムを実装する。スマートフォンを用いて IoT デバイスに Wi-Fi 設定を行うシステムのプロトタイプを図 8 に示す。図 8 は、システムのプロトタイプ実装の写真であり、左側が送信側であるスマートフォンで、右側は受信側である IoT デバイスが写っている。

送信側であるスマートフォンのアプリケーションは

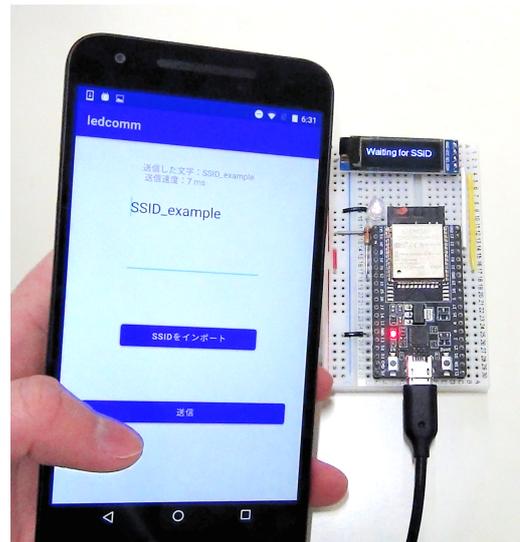


図 8 スマートフォンのフラッシュライトによる可視光通信を行うシステムのプロトタイプ実装

Android 端末で使用することを想定している。図 8 のアプリケーションの画面の説明を以下に示す。

- (1) ユーザは中央のテキストボックスに IoT デバイスへ設定する情報を入力する。初期設定情報を入力する方法として以下の 2 つがある。
 - (a) SSID やパスワードを直接入力する
 - (b) 画面中央の「SSID をインポート」ボタンをタップし、現在接続している Wi-Fi の SSID を自動で入力する
- (2) スマートフォンの画面下部にある「送信」ボタンをタップすると、フラッシュライトを点滅させて入力データを光信号として送信する。

受信側である IoT デバイスの実装は、インタフェースが LED のみの場合を想定し、LED 1 つでデータの受信と現在の IoT デバイスの状態が分かるようにした。図 8 の IoT デバイスの説明を以下に示す。

 - (1) プロトタイプ実装では、デバッグ用に IoT デバイス現在の状態をわかりやすくするために小型のディスプレイを接続した。
 - (2) 受信側の LED を電圧値の読み取りだけに用いるのではなく、IoT デバイスのステータスランプの役割をもたせ、1 つの LED を入力と出力を行えるインタフェースとして実装した。IoT デバイスがデータ受信準備状態になると LED が点滅し、Wi-Fi 接続完了時に LED が点灯するため、LED を見ることで IoT デバイスの状態を確認できる。

5. 関連研究

スマートフォンを用いて IoT デバイスの Wi-Fi を設定する技術として、ESP-TOUCH[6] がある。これは、Airkiss というプロトコルを利用して、IoT デバイスで Wi-Fi の設

定を行うものである。この技術を用いることで、スマートフォンにSSIDとパスワードを入力するだけでIoTデバイスへのWi-Fi設定ができるため、初期導入時の手間を減らせる。しかし、他のWi-Fi機器から初期設定の内容を傍受可能なためセキュリティの問題がある。一方、本手法では、通信は送信側と受信側を近づけて通信を行うため、Wi-Fiの設定を傍受される危険性が低い。

LEDを高度なインタフェースにする技術として、LightAnchor[7]とInfoLED[8]がある。どちらの技術も、LEDを点滅させることで情報を発信し、スマートフォンのカメラでLEDの光を撮影すると、そのLEDの点滅に含まれている情報をARで表示することを目的としている。これにより、スマートフォンでIoTデバイスと双方向の通信が可能となり、IoTデバイスの初期設定だけでなく、IoTデバイスの管理も可能になると考える。

6. おわりに

本論文では、スマートフォンのフラッシュライトによる可視光通信手法の検討と、その手法を応用しIoTデバイスの初期設定を行うシステムの設計を行った。

次に、スマートフォンのフラッシュライトを用いて可視光通信を行えるかを知るために、フラッシュライトの制御精度と光の強さの観点からフラッシュライトの特性調査を行った。調査の結果、スマートフォンでフラッシュライトを用いた可視光通信は可能だが、スマートフォンとIoTデバイスとの距離を離して複数台同時に設定することは、光の強さの観点から難しいことが分かった。調査結果からプロトコルの改良を行い、リーダ部を伸ばすことでズレが通信精度に与える影響を小さくした。プロトコル改良によって、通信精度を既存手法よりも最大31ポイント向上させた。

最後に、スマートフォンのフラッシュライトによる可視光通信手法の応用として、IoTデバイスの初期設定を行うシステムの設計及び実装を行った。

残された課題として、スマートフォンの光信号送信の安定化と、スマートフォンで光信号を受け取る機能の実装がある。

参考文献

- [1] Google Inc.: Google Home — Google Developers, Google Inc. (online), available from (<https://developers.google.com/home>) (accessed 2021-12-23).
- [2] 瀧本時空, 乃村能成: 可視光通信を利用したIoTデバイス設定手法の検討, 情報処理学会研究報告, Vol. 2020-DPS-184, No. 2, 第184回マルチメディア通信と分散処理・第89回電子化知的財産・社会基盤合同研究発表会, pp. 1-7 (2020).
- [3] Yang, Y., Luo, J., Hao, J. and Pan, S.: Counting via LED sensing: Inferring occupancy using lighting infrastructure, *Pervasive and Mobile Computing*, Vol. 45, pp. 35-54

- (2018).
- [4] Google Inc.: Android Developers, Google Inc. (online), available from ([https://developer.android.com/reference/java/lang/System.html#currentTimeMillis\(\)](https://developer.android.com/reference/java/lang/System.html#currentTimeMillis())) (accessed 2022-01-22).
- [5] 東山知彦, 増田大樹, 松本利夫: Androidのリアルタイム性評価及び改善方法の検討, 情報処理学会第74回全国大会講演論文集, Vol. 74, No. 3A-3, pp. 29-30 (2012).
- [6] EspressifSystems: ESP-TOUCH User Guide, Espressif Systems (online), available from (https://www.espressif.com/sites/default/files/documentation/esp-touch-user_guide_en.pdf) (accessed 2020-08-17).
- [7] Ahuja, K., Paredy, S., Xiao, R., Goel, M. and Harrison, C.: LightAnchors: Appropriating Point Lights for Spatially-Anchored Augmented Reality Interfaces, *UIST '19: Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pp. 189-196 (2019).
- [8] Yang, Jackie (Junrui) and Landay, James A.: InfoLED: Augmenting LED Indicator Lights for Device Positioning and Communication, *UIST '19, New York, NY, USA, Association for Computing Machinery*, p. 175-187 (online), available from (<https://doi.org/10.1145/3332165.3347954>) (2019).