

# 可視光通信を利用したIoTデバイス設定手法の検討

濱本 時空<sup>1</sup> 乃村 能成<sup>1</sup>

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。本稿では、IoTデバイスの設定において可視光通信の利用を検討する。この手法では、LEDの受光素子としての性質に着目し、LEDを利用した可視光通信を行うことで、価格上昇を抑えつつ初期設定の手間を削減する。また、実験を行い、提案手法の有用性の度合いを調査する。

キーワード：IoT, 可視光通信, マイクロコントローラ

## 1. はじめに

IoT (Internet of Things) とは、モノのインターネットと呼ばれ、今までインターネットに接続していなかったさまざまなモノがインターネットを通して情報を交換する仕組みである。IoTではIoTデバイスを用いてモノの状況や動作等膨大なデータをインターネットを介して収集することで、現在の状態を把握する。IoTデバイスを導入することで、例えば温度や湿度といったデータを自動で計測できるため、容易に大量のデータを集めることができる。

しかし、IoTデバイスの導入には初期設定が必要であることと、IoTデバイスの価格の問題がある。これらはIoTデバイス導入のしやすさに影響する。このため、初期設定を行いやすくしつつ、IoTデバイスの価格の上昇を抑える必要がある。例えば、データの収集をWi-Fiで行う場合、Wi-Fiを使用するための設定が必要となるが、Wi-FiのSSIDとパスワードをどのようにして設定するかが重要になる。IoTデバイスにPC等を有線で接続し初期設定専用の書込ソフトウェアを用意することで、IoTデバイスそのものの価格は抑えられるが、設定に必要な機材や前提知識が増大し、利用者の負担となってしまう。

そこで本稿では、多くのIoTデバイスに備わる可視光のLEDのみを用いてIoTの初期設定を試みた結果を報告する。

可視光通信自体は古くから研究開発が行われ、高速な可視光通信も技術として確立されている。たとえば、Li-Fi[1][2]は、照明等の光を可視光通信に用いる技術で、3.5Gbpsを

超える伝送速度を実現している。Li-Fiのように可視光通信専用の受光素子を低価格低機能なIoTデバイス求めるのは難しい。一方、LEDによる可視光通信であれば、もともと低価格低機能なIoTデバイスにも最低限のインタフェースとして実装されている。このLEDで可視光通信を行えば、IoTデバイスのハードウェア拡張を最小限にしつつ、データ通信を行える。

可視光LEDを利用した既存の双方向通信手法[3]では、通信速度は、約250bpsと報告されている。本提案で試みた手法では、5倍以上の通信速度が得られた。これにより、想定する環境において、概ね1秒以内という機器間の接触時間で初期設定を終えられることが分かった。

以降、本手法の通信方式を説明し、速度とデータの受信成功率の関係を調査し、妥当なフレーム構造を求めた過程を説明する。また、本手法を用いてIoTデバイスのWi-Fi設定を試みた結果を示す。

## 2. IoTデバイス導入時における問題点

### 2.1 IoTデバイス導入で想定されるシナリオ

IoTデバイス導入のシナリオとして、小学校で教室内の環境を管理する場合を考える。教室内の温度、湿度、および二酸化炭素濃度といった物理的環境は、生徒の学習効率に影響する。このため、温度の調節や、適度な換気等を行うことは生徒にとって有益である。そこで、これらの環境の監視をIoTデバイスを用いて行う。このIoTデバイスを、教室ごとに1つずつ設置し、センサで計測したデータをWi-Fiを介して収集する。

しかし、このIoTデバイス導入には、2つ問題がある。1つ目は、IoTデバイスに初期設定が必要なことである。IoT

<sup>1</sup> 岡山大学大学院自然科学研究科  
Graduate School of Natural Science and Technology,  
Okayama University

デバイスを利用するために1つ1つに初期設定を行う必要があり、導入者の手間になる。2つ目は、IoT デバイスの価格にある。初期設定を行いやすくするために高機能化すると、価格が上昇する。一方で、価格を抑えると機能が制限され、初期設定の手法も限られる。

## 2.2 IoT デバイス導入で想定される問題点

2.1 節のシナリオで述べたように、IoT デバイスの導入には以下の2つの問題がある。

### 問題1 初期設定の手間が大きい

IoT デバイスの種類や導入方法はさまざまであるが、IoT デバイスは導入時に初期設定が必要となる。このユースケース例のIoT デバイスではネットワークに接続する必要がある。初期設定の内容として、Wi-Fi を使用するためのSSID やパスワード、データ送信先の情報等が挙げられる。Wi-Fi の初期設定ではSSID とパスワードの最長文字数はそれぞれ32字と64字であるため、最長96文字分のデータを設定する必要がある。これらの設定を各IoT デバイス毎に行わなければならない、初期設定に手間がかかる。また、有線接続の場合、初期設定を行うために多少IT分野の専門知識が必要となる。しかし、全員が専門知識を持っているとは限らず、設定できる人が限られる。このため、少数の者がすべてのIoT デバイスの初期設定を担当する、という事態が起こりえる。

### 問題2 IoT デバイスの価格が高い

IoT デバイスの機能を増やすことで、初期設定方法や通信方法をより簡略化できる。例えば、IoT デバイスのインターフェースとしてタッチスクリーンを付属させて直接操作することでIoT デバイスの設定を行う、通信方法としてBluetooth を使用し初期設定を行う、といったことが考えられる。しかし、IoT デバイスの機能を増やすことは、構成部品が増えるほか、IoT デバイスのハードウェアが複雑となり、IoT デバイス本体の価格を上昇させることにつながる。一方で、インターフェースや通信方法を最小限にすると、構成部品が少なくなり回路も単純になるため、価格は安くなる。しかし、機能が制限されることにより設定方法が限られてしまう。もしインターフェースがUSBポートしかない場合は、初期設定時には計算機に有線で接続する必要があり、設定の手間が増えてしまう。また、インターフェースがない場合、IoT デバイスの稼働状況の判断が難しく、IoT デバイスの管理が難しくなる。

上記2つの問題は、どちらも導入するIoT デバイスの数に比例して増大する。シナリオ例において、教室が30室あった場合は、IoT デバイスを30個導入する必要がある。IoT デバイスの機能を増やし、誰でも簡単に設定できるよ

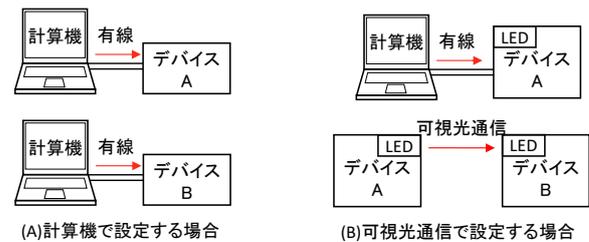


図1 IoT デバイスを計算機に有線でつなぎ設定する手法と可視光通信で設定する手法

うにした場合、スムーズに初期設定を行えるため、導入時の手間は減少する。しかし、機能追加によりIoT デバイスの価格が上昇する。一方で、低価格のIoT デバイスを用いた場合、IoT デバイス自体にかかる価格は減少する。しかし、インターフェースが低機能で初期設定に手間がかかり、設定できる人も限られるため、結果として導入時の手間が増加する。

このため、IoT デバイス導入時の初期設定の手間を削減しつつ、IoT デバイスの価格上昇を抑えるような仕組みが必要となる。

## 3. IoT デバイス導入時の問題への対処

2.2 節で述べたように、IoT デバイスの初期設定の手間を削減して、なおかつ価格上昇を抑える仕組みが必要となる。この問題の対処として、LED による可視光通信を利用したIoT デバイス同士による設定を検討する。

### 対処1 可視光通信による非接触の設定

有線接続による設定は、複数のIoT デバイスを設定する場合は繋ぎなおす手間がかかる。一方で、LED を利用した可視光通信での設定の場合、LED をかざすことで設定を行うため、繋ぎなおす手間がかからない。

### 対処2 LED を用いた回路の単純化

LED には、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う性質を持つ[4]。この性質より、1つのLED を、電流を流すことで可視光通信の送信部として用い、受光時の電圧を利用することで受信部として用いることができる。このため、1つのLED で可視光通信の送信部と受信部を担うことができる。多くのマイクロコントローラには、入力と出力を切り替えられる入出力端子が備わっている。このため、LED をマイクロコントローラの1つの端子に接続するだけで可視光通信が可能となる。また、可視光であるため、目視で通信が行われているか否かを確認できる他、LED の点滅パターンを変化させることでIoT デバイスのインターフェースとなり、機能性を著しく損なうことなく単純化できる。

この手法を用いた初期設定の例を図1に示す。図1は、IoT デバイスのデバイスAとデバイスBに対して初期設定を行う際に、それぞれ有線接続と可視光通信を利用した

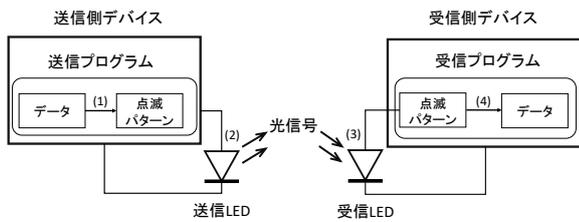


図 2 可視光通信を利用したデータ送受信における送信側と受信側の処理流れ

場合の違いを表している。図 1(A) は計算機で設定する場合を表しており、すべての IoT デバイスと有線をつなぐ手間が発生する。図 1(B) は可視光通信で設定する場合を表している。この手法では、最初の 1 台を有線で設定し、設定済みのデバイス A とデバイス B で可視光通信を行うことで、有線をつなぐことなく設定を行う。つまり、図 1(B) の手法では、設定が完了したデバイスを用いて初期設定が行えるようになるため、ねずみ算的に初期設定デバイスを増やせる。また、計算機に接続されたデバイス A を可視光通信のインタフェースとして計算機から制御することで、デバイス B, C といった他のデバイスに有線接続せずとも、デバイス A を介して通信が可能となる。このため、図 1(B) で、初期設定が必要な IoT デバイスが多い場合にかかる手間を大幅に削減できると考える。

4 章で可視光通信を利用したデータ送受信の手法を設計する。次に、設計した手法において、LED 可視光通信による通信速度が、想定したシナリオにおいて有用性があるのか分からない。このため、4.4 節で通信速度の計測を行い、4.5 節でその結果と本手法の有用性を示す。有用性の目標として、Wi-Fi の設定を想定した場合、SSID とパスワードの合計は最長 96 文字であるため、96 文字のデータを 1 秒以内に送信できることとする。最後に 5 章にてデータ送受信手法を用いた Wi-Fi 設定の実装を行い、利用方法についても述べる。

## 4. LED 可視光通信を利用したデータ送受信手法

### 4.1 概要

3 章で述べた対処を実現する手法として、LED を利用した可視光通信の設定手法を設計する。本手法では、1 つの LED を可視光通信の送信部と受信部として用いる。

可視光通信を利用したデータ送受信手法における送信側と受信側の処理流れを図 2 を用いて説明する。図 2 では、左側に送信側、右側に受信側の構成を示している。それぞれの IoT デバイスにはプログラムを保存しておき、プログラムによって IoT デバイスに接続された LED を制御する。システムの処理流れを以下に示す。

- (1) 送信プログラムで、送信したいデータを光信号の点滅パターンにエンコードする。

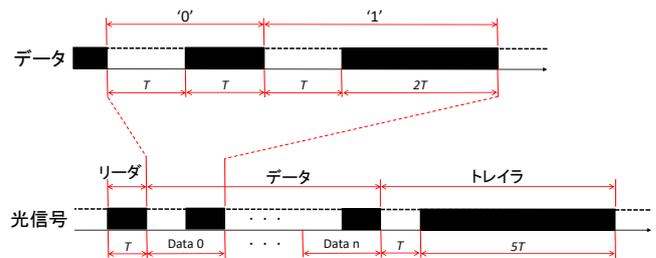


図 3 光信号のデータフレーム形式

- (2) 送信 LED を制御し、点滅パターンに合わせて送信 LED を点滅させ、光信号を送信する。
- (3) 受信 LED で光信号を受信し、光が当たっている時間から点滅パターンを読み取る。
- (4) 受信プログラムで点滅パターンから元のデータをデコードする。

### 4.2 光信号のデータフレーム形式

送信する光信号のデータフレーム形式を図 3 を用いて説明する。データフレーム形式について、図 3 下、「光信号」を用いて説明する。横軸は時間を表しており、黒く塗りつぶされた部分は LED の点灯を表している。まず、光信号の全体の構成を述べる。光信号には、データフレームのリーダー、データおよびトレイラが順に並べられている。リーダーでは光信号で基準となる時間の  $T$  時間だけ LED を点灯させる。その後、データで送信したいデータを送信する。このデータについて、図 3 上、「データ」を用いて説明する。横軸、縦軸、および黒く塗りつぶされた部分の意味は図 3 下、「光信号」と同じである。送信するバイナリデータの '0' と '1' を、LED の点灯時間の長さでそれぞれ表している。送信するデータが '0' の場合は、 $T$  時間の消灯の後に  $T$  時間だけ点灯させる。送信するデータが '1' の場合は、 $T$  時間の消灯の後に  $2T$  時間点灯させる。以上がデータの説明である。データの送信が終了したら、トレイラを送信する。トレイラでは、LED を  $T$  時間消灯した後に  $5T$  時間点灯する。これにより、データ部のデータをすべて送信し終えたことを受信側に伝える。

### 4.3 可視光通信によるデータ送受信の処理流れ

処理流れにおける、送信プログラム、受信プログラムおよび光信号の関係を図 4 を用いて説明する。横軸は時間を表しており、縦軸はプログラムの実行と停止を表している。送信プログラムと受信プログラムに振られた英数字はプログラムにおける処理を表している。送信プログラムの処理には通番の先頭に 's' を、受信プログラムの処理には通番の先頭に 'r' を付与している。処理の流れを以下に示す。また、受信プログラムの詳細な処理流れを図 5 のフローチャートに示す。

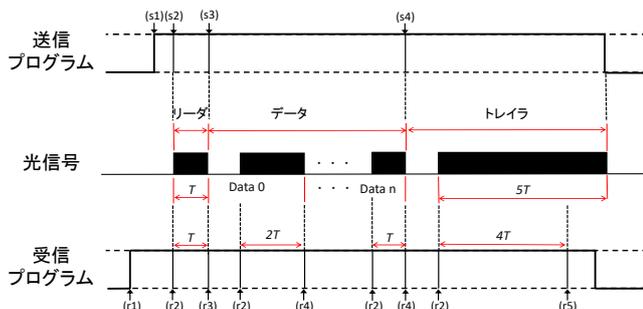


図 4 送信プログラム，受信プログラムおよび光信号の処理流れ

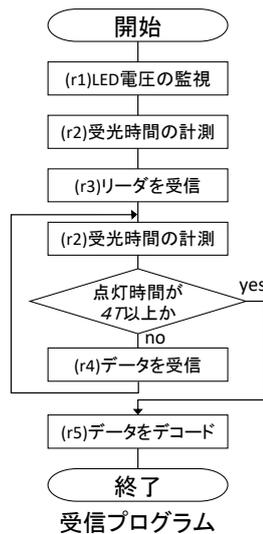


図 5 受信プログラムのフローチャート図

- (r1) 受信側で受信プログラムを実行し，LED の電圧値の監視を始める．
  - (s1) 送信側で送信プログラムを実行し，送信するデータを点滅パターンにエンコードする．
  - (s2) 送信側はリーダを送信する．
  - (r2) 受信側は，電圧値が閾値以上になった時間を計測し，受光時間を計測する．
  - (r3) 受信側は，受光時間からリーダの長さ  $T$  を取得する．
  - (s3) 送信側はデータを送信する．
  - (r4) 受信側は，リーダを受信した後，データを受信する．各データの信号の長さからそのデータが 0 か 1 かを判別する．
  - (s4) 送信側は，データをすべて送り終わるとトレイラを送信し，データをすべて送り終わったことを受信側に伝える． $5T$  時間 LED を点灯させた後，送信プログラムを終了する．
  - (r5) 受信側は，受光時間が  $4T$  時間を超えたため，データの読み取りを終了し，データのデコードを行う．
- 本手法では，可視光通信の通信速度は基準となる時間  $T$  の長さと同フレーム長に依存する． $T$  時間を短くすればスループットは向上する．あるいは，フレーム長を長くすれば転送効率が高まる．しかし，ビット誤り率やフレーム口

ス率が増加する．よって，妥当な基準時間  $T$  の長さと同フレーム長を実測により求める．

#### 4.4 実効スループットの計測

##### 4.4.1 概要

計測は，以下の 2 つの条件を変えて行い，実効スループットを計測する．

- (1) 基準時間  $T$
- (2) フレーム長

計測は以下の手順で行う．

- 手順 1 フレーム長に揃えた，0 と 1 のビットがランダムなデータを 300 回送信し，フレーム内すべてのデータを受信成功した確率を求める．
- 手順 2 手順 1 を，基準時間  $T$  とフレーム長を以降の 4.4.2 項と 4.4.3 項に述べる条件で変化させ，各条件でのフレーム受信成功率を求める．
- 手順 3 手順 2 で求めたフレーム受信成功率から実効スループットを導出する．

##### 4.4.2 基準時間 $T$

本実装における LED 可視光通信で送信する光信号の長さは基準時間  $T$  によって変動し， $T$  を変化させると通信速度が変化する．この時， $T$  を短くした場合，信号の長さは短くなりスループットは向上するが，ノイズの影響を受けやすくなりデータ受信の成功率が低くなると考える．そこで，基準時間  $T$  を  $200\mu\text{s}$  から  $1000\mu\text{s}$  の間で  $100\mu\text{s}$  単位で変化させ，実効スループットへの影響を計測する．なお，基準時間  $T$  が  $100\mu\text{s}$  の条件では可視光通信が正しく機能しなかった．

##### 4.4.3 フレーム長

連続してデータを受信する場合，ノイズにより受信のタイミングがずれると，ノイズ以降のデータ受信に影響を及ぼす．このため，データを一定の間隔で区切ることで，フレーム受信の成功率が向上すると考えられる．

そこで，フレーム長を 32bit から 2 倍ずつ大きくしていき，最大 1024bit にした時の精度としてフレーム内すべてのデータを受信成功した確率を計測する．送信側と受信側での同期を行うために，送信側はトレイラを送信後に 20ms 待機し，受信側はトレイラを受信後 15ms 待機する．

#### 4.5 計測結果

基準時間  $T$  とフレーム長を変化させた際のフレーム受信成功率の計測結果を表 1 に示す．また，表 1 をグラフ化したものを図 6 に示す．

縦軸がフレーム受信成功率，横軸がフレーム長，各凡例が基準時間  $T$  を表している．表 1 と図 6 を見ると，基準時間  $T$  が長くなるにつれ，また，フレーム長が短くなるにつ

表 1 基準時間  $T$  とフレーム長を変化させた際のフレーム受信成功率

基準時間 $T$	フレーム長					
	32bit	64bit	128bit	256bit	512bit	1024bit
200 $\mu$ s	0.990	0.967	0.943	0.883	0.837	0.747
300 $\mu$ s	0.990	0.983	0.990	0.957	0.910	0.907
400 $\mu$ s	0.990	0.980	1.000	0.973	0.907	0.833
500 $\mu$ s	1.000	0.990	0.987	0.970	0.923	0.920
600 $\mu$ s	0.997	0.993	0.993	0.973	0.937	0.880
700 $\mu$ s	0.993	0.993	0.980	0.973	0.930	0.920
800 $\mu$ s	0.990	1.000	0.997	0.970	0.950	0.870
900 $\mu$ s	0.993	0.987	0.990	0.977	0.953	0.830
1000 $\mu$ s	0.993	1.000	1.000	0.980	0.937	0.867

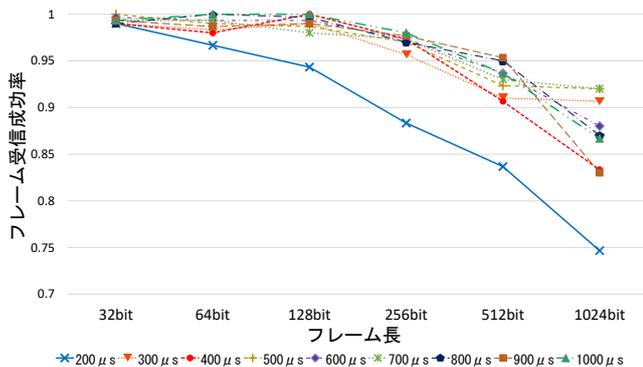


図 6 基準時間  $T$  とフレーム長を変化させた際のフレーム受信成功率の計測結果のグラフ

表 2 基準時間  $T$  とフレーム長を変化させた際の実効スループット (bps)

基準時間 $T$	フレーム長					
	32bit	64bit	128bit	256bit	512bit	1024bit
200 $\mu$ s	780.296	1034.56	1229.599	1292.19	1303.632	1202.558
300 $\mu$ s	622.397	789.628	922.942	969.927	964.837	983.815
400 $\mu$ s	517.647	629.719	725.624	755.071	728.521	681.794
500 $\mu$ s	447.552	530.209	586.048	609.374	597.279	604.091
600 $\mu$ s	389.894	456.05	499.398	513.759	507.056	482.553
700 $\mu$ s	345.132	399.079	427.102	442.975	432.833	433.08
800 $\mu$ s	309.375	357.143	383.333	388	387.755	358.763
900 $\mu$ s	282.047	317.161	340.737	348.469	346.494	304.511
1000 $\mu$ s	258.428	292.237	311.436	315.572	306.829	286.372

れてフレーム受信成功率が高くなっていることが分かる。この結果から各基準時間  $T$  とフレーム長の条件での実効スループットを求める。本実装において、すべてのビットが 1 のデータを送信した場合が一番遅くなるため、すべてのビットが 1 のデータを送信した際の実効スループットを求める。この結果を、表 2 と図 7 に示す。

表 2 と図 7 より、基準時間  $T$  が 200 $\mu$ s の条件で実効スループットが高くなっていることが分かる。また、フレーム長が短くなるとフレーム受信成功率は高くなっているが、フレーム間の待機時間の方がスループットに与える影響が大きいため、結果としてフレーム長が短くなるほど実効スループットが低くなっている。本実装で実効スループットが一番高くなるのは基準時間  $T$  が 200 $\mu$ s、フレーム長が 512bit の条件であることが分かった。この条件のスループットは約 1300bps であり、1 秒間に 162 文字ほど送信可能となる。

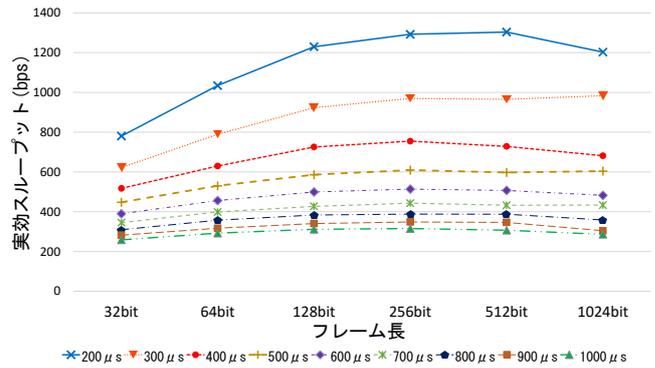


図 7 基準時間  $T$  とフレーム長を変化させた際の実効スループットのグラフ

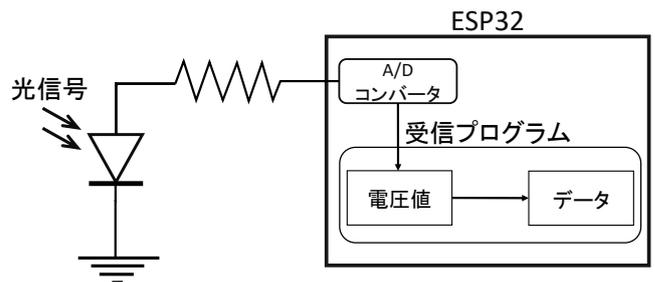


図 8 実装を行った受信側の回路図

## 5. データ送受信手法の実装と利用方法

### 5.1 手法の実装

4 章で述べたデータ送受信手法を用いて、Wi-Fi 設定を行うシステムのプロトタイプを実装した。送信側と受信側の実装は以下の通りである。

**送信側** 送信側 LED を Raspberry Pi に接続し、LED の点滅を制御する。

**受信側** マイクロコントローラの ESP32 を受信側の IoT デバイスとして、送信側 LED から Wi-Fi の SSID とパスワードを送信し、受信側でそれらのデータを受信し、データに基づいた Wi-Fi に接続する。接続が完了したら、受信で使用した LED を点灯させる。接続に失敗したら、LED を点滅させる。

実装した回路図を図 8 に示す。本実装では、受光側の LED に光が当たっている時間を計測するために、A/D コンバータを使用して LED の電圧を数値化し、電圧値が閾値以上の時間を計測している。ESP32 の A/D コンバータが利用できる入出力ピンに LED と抵抗を接続した。LED のアノードを入出力ピン、カソードをグラウンドに接続している。送信側 LED と受信側 LED は密着させて通信を行った。

本実装にて、送信側で Wi-Fi の SSID とパスワードを光信号に変換し送信した。受信側の LED で光信号を受信し、エンコードすることで光信号からデータへ復元できることを確認した。また、復元したデータを用いて Wi-Fi に接続

した後に、受光側 LED を点灯できることを確認した。

## 5.2 利用方法

本実装の利用方法として、IoT デバイスの LED を近づけて、ボタン 1 つで Wi-Fi の設定を行うことが考えられる。以下に Wi-Fi の設定を行う際の利用手順を示す。

- (1) 設定を行いたい IoT デバイスの電源を入れる。
- (2) 起動後、光信号を受信できるようになると、IoT デバイスの LED が短い間隔で点滅を繰り返す。
- (3) 既に Wi-Fi の設定を完了した IoT デバイスを用意し、設定済みの IoT デバイスの LED と、未設定の IoT デバイスの LED を近づける。
- (4) 設定済み IoT デバイスに付属しているボタンを短く押すと、光信号の送信を開始する。未設定の IoT デバイスの LED でこの光信号を受信する。
- (5) 光信号を送信後、未設定の IoT デバイスの LED で設定が完了したか確認する。LED が点灯したままだと、Wi-Fi の設定と接続が完了している。また、LED が点滅していると、Wi-Fi の接続に失敗しているため、再度 (4) からの手順を行う。
- (6) まだ未設定の IoT デバイスがある場合、再度 (1) からの手順を行う。

この利用方法では、ボタン 1 つで設定が完了するため、有線接続の場合と比べて設定時の手間が少ない。また、IoT デバイスの Wi-Fi を可視光通信で設定した後に、IoT デバイスの状態を LED を見て確認できるため、設定に成功したのか失敗したのかが分かりやすい。さらに、設定し終わった IoT デバイスを手順 (3) の設定済み IoT デバイスとして利用するため、すべての IoT デバイスを設定しきるまでの時間を短縮できる。

## 6. 関連研究

本稿では A/D コンバータを使用して電圧値を計測することで、LED による光検出を行った。文献 [3] では点灯および消灯を電圧値と閾値で判別するのではなく、光が強くなると LED の内部容量の放電速度が速くなる性質を利用して、放電速度の差で判別している。こちらの方が、A/D コンバータを使用しなくとも光検出を行えるため実装の幅が広がるが、光検出を行うたびに LED に充電が必要になるため、より速い通信速度を実現しようとした場合にボトルネックとなると考える。

LED の受光素子としての性質に着目した研究として、LED を利用した低コストかつ双方向の通信を行う手法 [3] がある。文献 [3] では、マイクロコントローラに LED を 1 つ取り付け、約 250bps の通信速度で双方向の可視光通信を実現している。この通信速度では、96 文字のデータを送信するためには約 3 秒かかる。本稿では、IoT デバイスの

初期設定を IoT デバイスとして ESP32 を使い、約 1300bps の通信速度で可視光通信を実装した。本実装では、96 文字のデータを 1 秒以内で送信できる。

IoT デバイスの Wi-Fi を設定する技術として、ESP-TOUCH[5] 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。設定手順を以下に示す。

- (1) 既に Wi-Fi に接続しているスマートフォンから、アクセスポイントに特定の packets を送る。この packet 長に意味を持たせることで、Wi-Fi の SSID とパスワードを表現している。
- (2) IoT デバイスがその packet 長を監視し、packet 長から SSID とパスワードをデコードする。
- (3) IoT デバイスはデコードした情報をもとに、Wi-Fi に接続する。

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、設定には専用のアプリケーションが必要である。また、他の Wi-Fi 機器から傍受可能なためセキュリティの問題もある。一方、本稿では、通信は LED 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[6] と InfoLED[7] 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、インタフェースが貧弱な IoT デバイスでも、この技術を利用することでさまざまな情報を発信できる。

## 7. おわりに

本稿では IoT デバイスの初期設定として、データ送受信手法を提案した。データ送受信手法は、LED の受光素子としての性質に着目し、1 つの LED で光信号の送信と受信を行う。送信側の LED を点滅させて光信号を送信し、受信側の LED でその光を受光した際に生じる電圧から、受光時間を計測する。受光時間から光信号を読み取り、データを復元する。

次に本手法の通信速度とフレーム長を変化させた際のフレーム受信成功率を計測し、約 1300bps の実効スループットを確認した。これは、1 秒間に 162 文字程度送信できるため、IoT デバイスに数十文字程度のデータを設定する場合、本手法は有用であるといえる。

最後に本手法による Wi-Fi 設定を行うシステムの実装と、その利用方法を述べた。利用方法では、Wi-Fi 設定済みの IoT デバイスを用いて、未設定の IoT デバイスへボタン 1 つで設定する流れを示し、本手法の有用性を示した。

## 参考文献

- [1] LiFi.co: LiFi: Wireless data from every light bulb, LiFi.co (online), available from <https://lifi.co/> (accessed 2020-08-08).
- [2] Sarkar, A., Agarwal, S. and Nath, A.: Li-Fi Technology: Data Transmission through Visible Light, *IJARCSMS*, Vol. 3, pp. 1–10 (2015).
- [3] Paul, D., William, Y. and Darren, L.: Very Low-Cost Sensing and Communication Using Bidirectional LEDs, *UbiComp 2003: Ubiquitous Computing* (Dey, Anind K. and Schmidt, Albrecht and McCarthy, Joseph F., ed.), Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 175–191 (2003).
- [4] 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).
- [5] EspressifSystems: ESP-TOUCH User Guide, Espressif Systems (online), available from [https://www.espressif.com/sites/default/files/documentation/esp-touch\\_user\\_guide\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp-touch_user_guide_en.pdf) (accessed 2020-08-17).
- [6] 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).
- [7] 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. 175187 (online), available from <https://doi.org/10.1145/3332165.3347954> (2019).