

# Webmo: Wifi と WebAPI をパッケージングしたステッピングモーター

原 健太\*      渡邊 恵太†

**概要.** 実世界と直接関わるシステムを試作する場合には Arduino などのツールキットを用いることで高速に開発できるが、自分の生活内で活用するために実際に長時間使用したり、他のシステムと連携させるためにネットワークに参加させたりしたい状況は少なくない。しかし、これらの実装にはネットワークやハードウェアの知識など、幅広い知識が要求され、実装に時間もかかる。高速に試作を繰り返したい場合はこうした工程をスキップして、アプリケーションのコーディングやユーザーとのインタラクション設計に集中できることが理想であるが、この考えのもと作成されたアクチュエータは存在しない。そこで本論文では、すぐに試作に取り掛かることのできる Web API を備えたステッピングモーター、Webmo を試作したので紹介する。

## 1 はじめに

実世界と直接関わるシステムを試作する場合には Phidgets[1]や Arduino, Raspberry Pi などのツールキットを用いることで高速に開発できるが、生活内で活用するために実際に長時間使用したり、他のシステムと連携させるためにネットワークに参加させたりしたい状況がある。こうした状況では、デバイスをコンピュータに繋ぎっぱなししておくことは現実的ではない。デバイスによっては、スタンドアロン化したり、ネットワークに参加させるためにモジュールを新たに追加したりする必要がある。ネットワーク経由でのデータ取得やアクチュエータ操作のために必要なステップは次のようになる。

1. ツールキットでセンサの値を取得したり、アクチュエータを操作したりできるようにする
2. ツールキットがネットワークに参加できる準備をする
3. ツールキットに通信してデータをやり取りするためのプログラムを記述する
4. ツールキットと他のシステムをつなげるためのプログラムを両方もしくはどちらかに記述する

このように、ただネットワーク経由でセンサの値を取得したりアクチュエータを制御したりしたいと考えても、数多くの工程を踏む必要があり手軽とは言えない。また、自分で作業しなければならない層が多いので、広い知識と経験が必要で、問題の切り分けも難しい。実際に製品製作をしたり、小型化したりする場合には Arduino などの小さなまとまりのモジュールを組み合わせれば良いが、高速に

Copyright is held by the author(s).

\* 明治大学, † 明治大学/JST

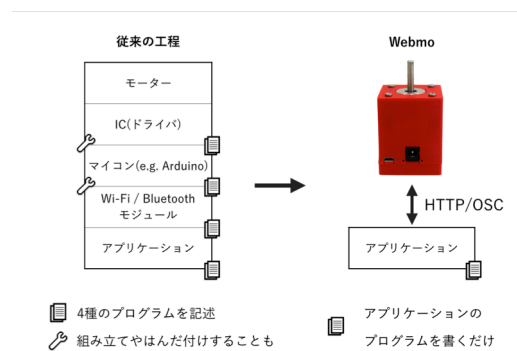


図 1. 現状との比較と、Webmo が解決する範囲

試作を繰り返したりしたい場合には、先述した工程の 1~3 に該当するハードウェア内部の作業を省き、アプリケーションのプログラムや、インタラクション設計に集中できることが理想であると考えられる。

こうした問題に対して、Tesselや Particle Photon は、ツールキットに Wi-Fi モジュールを組み合わせることですぐにボードをネットワークに参加させようとしているが、ツールキット上で外部とやり取りするコードや、センサを制御するコードを記述する必要がある。塚田らの MobiServer[2]はネットワーク経由で DMX 機器や Phidgets の制御を簡単に行えるようにしたが、デバイスがコンピュータに接続されていることが前提となっている。Texas Instruments 社の Sensor Tag はセンサに Bluetooth などの無線通信モジュールを組み込み、電子工作の知識がない人でもすぐにアプリケーションの試作に取り組めるようにした。しかし、電子工作の知識が無くとも、すぐに試作に取り掛かれるアクチュエータは存在していない。そこで、Web API を持ったステッピングモーターWebmo を試作したので紹介する。図1に解決する問題をまとめた。

ソースコード 1. モーターを 3 秒間回転させる

```
1 var webmo = new Webmo("Webmo.local");
2 webmo.rotate(60);
3 setTimeout(() => { webmo.stop(); }, 3000);
```

## 2 提案システム

Webmo は本体のハードウェア部分と、本体のソフトウェア部分、Webmo を操作するためのライブラリの 3 つで構成される。

ハードウェアはステッピングモーター (SM-42BYG011, MERCURY MOTOR 社)に加えて、モーターを動作させるためのモータードライバ (L6470, STMicroelectronics 社), ドライバを動作させるためのメインチップ (Edison, Intel 社)で実装した。チップや周辺の電源回路をすっきりとまとめてステッピングモーターの下部に集約させるため、自作の基板を作成している。基板には AC アダプタと USB Micro A アダプタの 2 つのみが接続できるようになっているが、基本的にユーザーが USB コネクタを利用することは無い。

Webmo 本体のソフトウェアは Node.JS で実装した。OS が立ち上がると Web サーバーが起動するようになっており、この Web サーバーが Web API として機能する。また Open Sound Control(OSC)のサーバーも同時に起動しており、HTTP の他にも OSC で操作することも可能である。

起動時に接続できるアクセスポイントが無ければ、Webmo は自分自身がアクセスポイントになるようにする。ユーザーは Webmo のアクセスポイントにアクセスし、Web ブラウザ経由で無線 LAN の接続設定を行うことができる。Webmo はマルチキャスト DNS で「Webmo.local」といったような自分のホスト名をネットワーク上に配信するので、ユーザーは Webmo 本体の IP アドレスを知らなくとも、Webmo に接続することができる。

Webmo を操作するためのライブラリは、現在 Processing と JavaScript 上で動作する二種類のもので実装されている。例として、JavaScript で 3 秒間、毎秒 60 度の速さで回転させるスクリプトをソースコード 1 に示す。ライブラリは Webmo の探索まで行うので、ユーザーは Webmo のホスト名のみ認識していれば、TCP/IP を意識せずに Webmo を利用することができる。

## 3 活用例

Webmo とスライドレール、タイミングベルト、カメラマウンタ、レーザーカッターで切り出したアクリルなどを組み合わせ、無線制御可能なカメラレ



図 2. 試作したカメラレール

ールを試作した。プログラムを JavaScript で作成すれば、スマートフォンなどから操作できるようなシステムを簡単に作成できる。

## 4 まとめと今後の課題

簡単にネットワーク経由で制御することが可能なモーター、Webmo を試作した。現在は 1 台の利用のみ想定しているので問題が無いが、複数台制御したい場合にはある程度の精度で同期を行う必要がある。したがって今後は複数台の Webmo の同時制御にも取り組む予定である。また、アクチュエータの制御に慣れないユーザーにとっては、思い描く動きに向けてモーターの回転をどのように変換していくのか、モーターの先に何を取り付けて、どのように回転運動を変換させていけば良いのかなどが難しく、そうしたレベルまでの支援を行う必要があるのではないかと考えている。今後は本システムを用いたハッカソンなどを通して、実際に Webmo を用いた試作を繰り返してシステムを洗練させると共に、モーターの動きのプログラミング方法などについても検討していきたい。

## 謝辞

ハードウェア部分に関して、神山洋一氏と鈴木涼太氏に協力を得た。

## 参考文献

- [1] S. Greenberg and C. Fitchett. Phidgets: easy development of physical interfaces through physical widgets. In Proc. of UIST '01, pp. 209–218, New York, NY, USA, 2001. ACM.
- [2] **MobiServer**: 日曜ユビキタスのための手軽なミドルウェア群 (2015/10/16 確認)  
<http://mobiquitous.com/mobiserver/>