

## 軽量言語による組み込みシステム開発環境の開発

金森 一真†

早川 栄一‡

† 拓殖大学 大学院 工学研究科

‡ 拓殖大学 工学部 情報工学科

### 1. 研究の背景

機器同士が人間を介さずに機器を制御し様々なサービスを提供する **Machine to Machine**[1][2] (以下 **M2M**) サービスが注目されている。最近では安価で高性能なマイコンを入手でき、個人レベルで **M2M** のようなシステムを構築できるようになってきている。しかし、組み込み開発の経験がない開発者がセンシング、サーバの構築、データの可視化のすべてを開発するのは難しい。また、センサの周囲環境をセンシングする機器 (デバイス) のハードウェア制御には **C** 言語が用いられているが、機能実装のためのコードが長くなり、デバッグが難しい場合がある。

このような問題に対して、センシング、サーバの構築、データの可視化アプリケーションを軽量言語で開発できる開発環境を提案する。本研究は組み込み開発の経験がないもしくは少ない開発者が小規模かつコードの記述が短い軽量言語で組み込み開発が行える環境を提供する。

### 2. 目的

本研究は軽量言語による組み込みシステム開発環境の開発である。具体的には統一の軽量言語でセンシング、サーバの構築、データの可視化が開発できることで、**C** 言語による組み込みシステム開発に比べ、簡潔なコードで複数の言語を学習することなく機能の実装を行える開発環境を開発者に提供することである。

本開発環境では、組み込み開発の経験が少ない開発者を対象とし、ホームセンシングのような小規模開発を想定している。開発経験が少ない開発者にとっては、複雑なコードを記述しなくてもサーバ記述と同じ言語で機能の実装を行うことができる。

Development of embedded system development environment with lightweight language

†Kazuma Kanamori

‡Eiichi Hayakawa

††Takushoku University, Electronics and Information Engineering

### 3. 問題分析

#### 3.1. 現在の組み込み開発についての問題点

センサ等のハードウェア制御には **C** 言語が用いられることが多い。しかし **C** 言語には1で述べたような問題がある。また、言語仕様が低水準であり、実行状態の把握が難しいことから、デバッグやライブラリ、ツールの開発が難しい。

#### 3.2. 要求仕様

軽量言語による組み込みシステム開発環境の要求を次に示す。

- (1) サーバと同じ言語で開発できる環境
- (2) データの可視化を容易にできる環境

### 4. 開発支援の方針

#### 4.1. 開発支援の方針

本開発環境を開発するにあたって重要なことは、簡潔なコードで機能の実現が可能でデータの可視化を容易にできることが必要である。特にデータの可視化はセンシングの目的であり重要な機能である。

これらの理由から本開発環境ではデータの可視化を容易に行えることを重点とし開発を行うものとする。

#### 4.2. 軽量言語による開発環境の設計方針

先述したように組み込み開発の経験が少ない開発者にとってセンシング、サーバの構築、データの可視化を **C** 言語で開発するのが難しいというのがある。これらの問題要素に対して本開発環境では次の開発環境方針を提案する。

##### (1) 軽量言語による開発

コードが簡潔に記述できることは、開発者の学習コストが少なく短期間でシステム開発が行える。特に軽量言語は少ない記述量で開発手順が簡略化されているため、開発者は容易にシステムを実現できると考えられる。

##### (2) データの可視化

データの可視化の際に開発者自身が描画処

理を一から実装するのはかなりの負担になると考えられる。描画処理はテンプレートを用意し開発者は可視化したいデータだけを描画できるようにする。それによって開発者の負担を減らせる。

## 5. 開発環境の設計

環境を開発するに際し、先に述べた方針を基に設計を行った。本環境ではシステム開発のための基本的な部分を自動生成する。自動生成することで開発者はアプリケーションの開発に集中することができる。本開発環境では JavaScript を用いて開発を行う。

### (1) ハードウェア制御

センサ等のハードウェア制御を JavaScript ベースで記述するために BoneScript[4]ライブラリを利用する。このライブラリを利用することによって BeagleBone Black の LED や GPIO の制御を行う。

### (2) サーバの構築

サーバの構築には Node.js[3]を利用する。Node.js はサーバーサイド実装の言語であり JavaScript ベースでサーバを構築することができる。

### (3) データの可視化

データの可視化には JavaScript ライブラリの D3.js[5]を利用する。D3.js は任意のデータを DOM と結合させデータ駆動によるドキュメントの変更を可能になる。

## 6. 実装

### 6.1. 実装環境

マイコンボードには図 1 の BeagleBone Black を使用しセンサには温度センサ、湿度センサ、照度センサの 3 つを使用した。

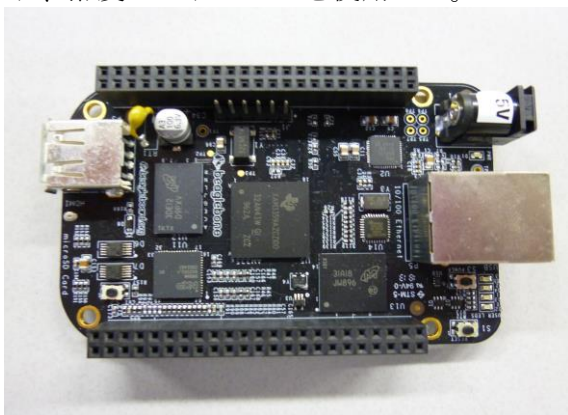


図 1 Beagle Bone Black

これを用いて、天気状態を監視するデバイ

スを開発した。各センサからのデータを可視化したグラフを示す。(図 2)

### 6.2. グラフの描画

グラフは Web ブラウザ上で見るができるようになっている。グラフの描画に D3.js を利用した。また、Ajax を用いることでリアルタイムなグラフを描画することができる。今回グラフの描画には折れ線グラフを用いて、周囲の状況の変化がすぐに把握できるようにした。

データをグラフに描画する際に、画面右上の設定ボタンを押すことで表示させたいデータを選択できるようになっている。

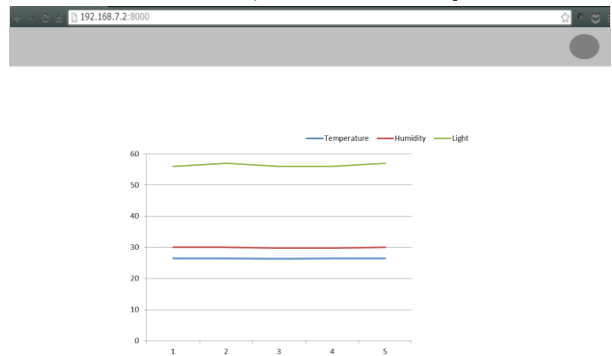


図 2 データ可視化グラフ

## 7. まとめ

軽量言語による組込みシステム開発環境の開発を行い、これにより軽量言語を用いてセンシング、サーバの構築、データの可視化を容易に行えることが可能となった。

今後の課題は、データの可視化に用いるグラフの追加とデバッグ機能の実装を行ってきたい。

## 参考文献

- [1] 清尾克彦「M2M (Machine to Machine) 技術の動向と応用事例」: [http://www.cyber-u.ac.jp/bulletin/0005/pdf/0005\\_02.pdf](http://www.cyber-u.ac.jp/bulletin/0005/pdf/0005_02.pdf)
- [2] 藤田隆史 後藤良則 小池新「M2M アーキテクチャと技術課題」: [http://www.ieice.org/jpn/books/kaishiki\\_ji/2013/201305.pdf](http://www.ieice.org/jpn/books/kaishiki_ji/2013/201305.pdf)
- [3] Node.js: [http://nodejs.jp/nodejs.org\\_ja/](http://nodejs.jp/nodejs.org_ja/)
- [4] BoneScript: <http://beagleboard.org/Support/BoneScript/>
- [5] D3.js: <http://ja.d3js.node.ws/>