

PCSS : Web 開発者のための WoT デバイス開発環境の提案

赤塚 大典^{†1} 山岡 潤一^{†2} 川井 敏昌^{†3}

概要: PCSS (Physical CSS) は Web 開発者のための WoT (Web of Things) デバイス開発環境である。Web 技術の一つである CSS でローレベルデバイス制御を可能とし、特にプログラミングにおいて Web ページ開発と同じように WoT デバイス制作が可能となる。開発の敷居を大幅に下げ且つ開発効率上がるだけでなく、Web サービスのマッシュアップデバイスやデバイスによる Web サービスの構築など、新しい Web Centric なデバイスの考案と開発を容易にする環境を提供する。

PCSS: WoT Device Development Environment for Web Developers

DAISUKE AKATSUKA^{†1} JUNICHI YAMAOKA^{†2} TOSHIMASA KAWAI^{†3}

Abstract:

PCSS (Physical CSS) is a WoT (Web of Things) device development environment designed for Web developers. It makes it possible to control low level hardware from CSS which is one of a Web technology. It means that developers can develop WoT devices in an environment which is similar to Web page development. It eliminates difficulties in WoT device development in existing methods. Also PCSS provides an environment for developers to develop new Web Centric Devices, such as a device mashed up with Web services or Web services based on devices, easier.

1. 背景

IoT (Internet of Things) が現在注目を集めている。日本においてはモノのインターネットと呼ばれ、コンピュータをはじめとした情報機器のみならず、全てのモノをインターネットでつなぐことで遠隔操作やセンシングなどを可能とし、こうしたインフラの中から新しいサービスを見出そうとしている。家電から自動車など利用者の利便性の向上や、工場や農業の効率化にいたるまで IoT デバイスが提案され実運用も行われている。

他方、WoT (Web of Things(i)) という考え方もその拡張として登場している。Web は言うまでもなく、業務から家事、エンターテインメントにいたるまで我々の生活のあらゆる場面に深く関わり、今後もその重要性が増すことが予想される。こうした背景から、WoT デバイスとは Web とデバイスが密に直結・連携し、むしろ Web や Web サービスを中心とした Web Centric Device な考え方を持つデバイスが今後提案されていくものと考えられる。これは単に Web につながるだけでなく、Web の考え方や作法、開発手法を持ち込むことで得られる新しい価値や体験を持つデバイスである。こうしたデバイスの実現には、いかに Web 開発者がデバイス制作に携われるかが鍵となる。

2. 先行事例

最近では、JavaScript でセンサやアクチュエータを制御できる開発ボードも登場してきており、Web 開発者によるデバイス制作の機会も増えた。Espruino(ii)や Tessel(iii)、Raspberry PI(iv)をはじめ、Arduino(v) も実行可能対象とするライブラリ Jonny Five(vi) なども登場し開発の敷居が下がった。しかしながら、たとえ JavaScript で制御できたとしても、ローレベルデバイスの制御プログラミングは Web 開発者にとっては難しい。例えば、I2C を介したアクチュエータの制御を考えてみても、指定したアドレスにコマンドを送信、数ミリ秒待ち、次のコマンドを送信など、デバイスに依存した操作は煩雑である。

3. PCSS (Physical CSS)



図 1 PCSS

3.1 設計

PCSS は Web 技術の一つである CSS でローレベルデ

^{†1}Mozilla Factory
Mozilla Factory

^{†2}慶応義塾大学 SFC 研究所 / 日本学術振興会
Keio Research Institute at SFC / the Japan Society for the Promotion of Science

^{†3}FabCafe LLP
FabCafe LLP

バイス制御を可能とする環境である。ローレベルデバイス制御ロジックを極力排除し CSS で代用することで、Web 開発者は Web ページ制作と同じように DOM と CSS の管理に注力すればよく、特にプログラミング開発の敷居を大幅に下げる。

CSS とは HTML エlementをどのように修飾するかを指定できる技術で、CSS の中にはローレベルデバイス制御にも適したプロパティが存在する。たとえば、文字色を指定する color は、フルカラー LED やランプなど文字通り色の変更が可能な制御に適しているし、HTML エlementの回転を指定する transform: rotate はサーボモータなどに利用できる。

敷居を下げる以外の利点は次の通りである。プロトタイプデバイス開発時には正しく動かないことがままあるが、その原因がソフトウェアにあるのかハードウェアにあるのか特定が難しい場合がある。PCSS では出力結果が Web UI にも反映されるので、少なくとも画面上で正しく動いているならば原因をハードウェアに絞って開発を進めることができる。Web UI を見ながらの開発は、ハードウェアが手元に無い状態においてもプログラミングが可能であるという点、その上で共同開発が容易になることや Web のテストツールが利用可能となる。

3.2 実装

本システムは、MozOpenHard(vii) コミュニティで試作を進めている WoT プロトタイプデバイス開発環境 CHIRIMEN を使用している。B2G という OSS 版 Firefox OS が搭載されており、レンダリングエンジン Gecko 上に JavaScript から GPIO および I2C を制御できる WebGPIO(viii), WebI2C(ix) という同コミュニティが W3C に提案の準備をしている(x) API を実装している。WebAPI の一つで DOM 内の変更を通知する機能を提供する MutationObserver を使って CSS によるスタイル変更を検知、WebGPIO, WebI2C を介してデバイスを制御する。HTML element とローレベルデバイスとのマッピングおよび先のスタイル変更の検知、ローレベルデバイスの制御は CSSDeviceManager が担う。

現在は、フルカラー LED とサーボモータをサポートしている。PWM 制御のため PC9685 チップ搭載ボードを CHIRIMEN に I2C 接続にて拡張、これを制御する。

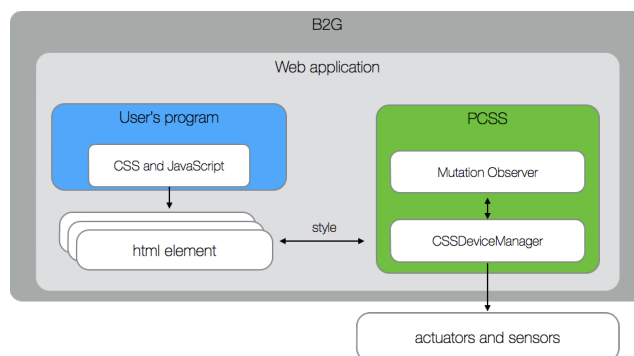


図 2 システム構成図

3.3 応用

PCSS を使ったプロトタイプをいくつか作成した。

Web Bear は twitter のつぶやきが反映される動くぬいぐるみである。予め決められたハッシュタグを持つツイートから :-) などの感情表現を取得、その表現にあわせてぬいぐるみが動きだす。感情表現が複数ある場合には、出現順に表現を行い、途中で感情表現以外の文字列がある場合には小休止するため、組み合わせにより簡単なぬいぐるみのアニメーションが可能となる。

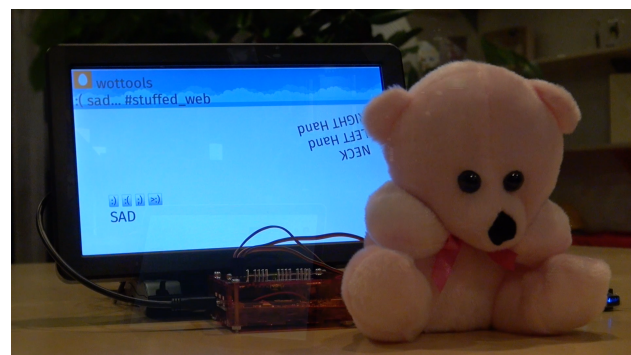


図 3 Web Bear

Calendar Lamp は Google カレンダーによりランプの色や光り方をスケジュール管理できるアプリケーションである。Google カレンダー上の現在時刻に予定されているスケジュールの色にあわせてランプの色が変化し、複数のスケジュールが登録されている場合はそれらの色を合算しランプ色に反映する。

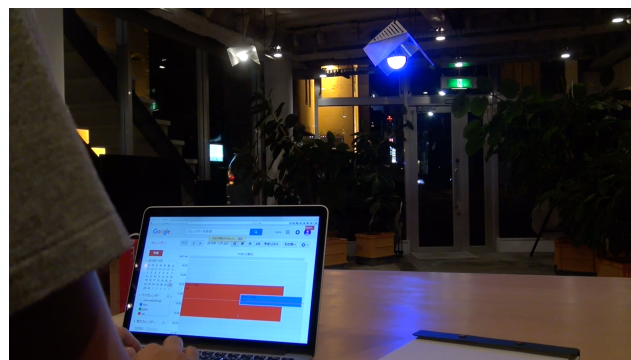


図 4 Calendar Lamp

ローレベルデバイス制御においても Web ページを制作すれば良いので通常の Web ページ同様、既存の Web サービスを取り込むことが簡単化され、Web サービスをマッシュアップしたデバイスなども視野に入る。

3.4 展示

Mozilla Festival 2015 という Mozilla 関係で比較的アカデミックおよび教育に関連したプロジェクトが一同に集まるイベントが 11 月 6-8 日にロンドンにて開催され、同イベントにてデモ展示を行った。また 11 月 15 日には Mozilla Developer Conference が東京にて開催されデモ展示した。デモ展示を見て頂いた方々からの主なフィードバックは以下のような内容であった。

- Web 開発者の私にとって、とても簡単に理解できる
- CSS が利用可能なので、エンジニア以外の例えば Web デザイナーにも使えそう
- Web とデバイスの距離感がとても近く感じられるようになった
- CSS の opacity でランプの明暗を制御できて面白そう
- CSS の transform: translate でモノを動かしてみたい
- 教育用途に使っていききたい
- 私も使ってみたい



図 5 Mozilla Festival 展示



図 6 Mozilla Developer Conference 展示

4. おわりに

本稿では、Web 開発者を対象とした WoT プロトotyping 開発環境 PCSS を扱った。Web 開発者の持つ技術によってデバイス制御が可能となり開発の敷居を大幅に下げる。デバイス制御に Web ページを作れば良いと考えれば、多様な Web サービスをマッシュアップしたデバイスや、デバイスそのものが Web サービスであるとの考えにも到達する。Web の実体としてデバイスが存在するという考えに近く、Web 開発者が作る Web Centric なデバイスが Web に新たな役割を与える。

本システムは当初 Web 開発者を対象としていたが、デバイス開発の初心者や学生など、広く教育現場においても利用可能である。ブラウザ Firefox の提供元である Mozilla は 読み・書き・Web を謳っており、今後我々が生活していく上での基礎体力として Web リテラシを持つべきだとしているが PCSS でのデバイス開発は、同時に Web 技術の習得にもつながる。

今後は取り扱えるアクチュエータやセンサを増やしていく、より多様で自由な Web Centric Device 構築のための開発環境を目指していきたい。

謝辞

MozOpenHard コミュニティのみなさまに感謝いたします。WebGPIO や WebI2C の考え方、またその実装がなければ、このアイデアも無かったと思います。

また Calendar Lamp はデジタルハリウッド大学大学院における「IoT/WoT プロトタイプ演習」授業でのアウトプットをフォークしています。デジタルハリウッド大学大学院の鈴木美香さま、萬里小路忠昭さま、河野功さま、王威さまに感謝いたします。

参考文献

- i Web of Things: <http://www.w3.org/WoT/>
- ii Espruino <http://www.espruino.com/>
- iii Tessel <https://tessel.io/>
- iv Raspberry PI <https://www.raspberrypi.org/>
- v Arduino <https://www.arduino.cc/>
- vi Johnny Five <http://johnny-five.io/>
- vii MozOpenHard <http://mozopenhard.mozillafactory.org/>
- viii WebGPIO <https://rawgit.com/browserobo/WebGPIO/master/index.html>
- ix WebI2C <https://rawgit.com/browserobo/WebI2C/master/index.html>
- x W3C Browser and Robotics Community Group <https://www.w3.org/community/browserobo/>