# Form Follows Function(): 形状と機能を単一コードで開発できる統合開発環境

加藤 淳[†]　　後藤 真孝[†]

**概要**．実世界で動作するデバイスを制作する際には，センサーやアクチュエーターなどを配置するためのレイアウトを CAD システムで設計し，マイコンのプログラムを統合開発環境で記述することが通例である．このような開発プロセスでは，両方のツールの使い方を習得する必要がある．また，各々のバージョンを同期しづらいだけでなく，ツール間で受け渡せない情報があり作業に無駄が生じる．本稿では，これらの問題を解決するため，実世界で動作するデバイスの形状（レイアウト）と機能（プログラム）を単一のコードベース（両義的ソースコード）で同時に指定できる統合開発環境 f3.js を提案する．

## 1　はじめに

実世界で動作するデバイスを制作するフィジカルコンピューティングや Internet of Things (IoT)の開発プロセスでは，レーザーカッターや 3D プリンタを用いてデバイスの筐体を出力し，センサーやアクチュエーター，マイコンを筐体に配置することが多い．近年，筐体の出力は高速かつ安価になり，マイコンの処理能力向上や統合開発環境技術の進展に伴い JavaScript のような動的言語でもマイコンのプログラミングが可能となった．

しかしながら，デバイスの形状を左右する筐体のレイアウトと，デバイスの機能を実装するマイコンのプログラミングには，CAD システムと統合開発環境という別々のツールが利用されている．本研究では，この開発プロセスにおける 3 つの問題に取り組む．まず，開発者は少なくとも 2 つのツールの使い方を習得する必要があり，敷居が高い．次に，各々のツールで生成されたファイル間の同期を取ることが容易ではなく，どの版のレイアウトとプログラムが対応づいているか混乱が生じかねない．最後に，ツール間の連携が不十分なため，作業に無駄が生じる．例えば，一部の CAD システムは余白の広さや利用モジュールの種類などを指定してパラメトリックにレイアウトを生成できる．これらのパラメタはプログラムの動作にも関わるが，CAD システムから開発環境に引き継ぐ方法がないため，プログラム開発において同内容を改めて入力する手間がかかる．

本稿では，上記の問題を解決するために，デバイスの**形状**（レイアウト）と**機能**（プログラム）を単一のコードベース（JavaScript のソースコード）で同時に指定できる統合開発環境 f3.js を提案する．



[†] 産業技術総合研究所

## 2　f3.js: Form Follows Function()

f3.js を用いると，ブラウザ上で JavaScript のソースコードを記述し，アクリル板を切断するためのレイアウトを出力してデバイスの物理的なインタフェースを設計できる．また，同一のソースコードを

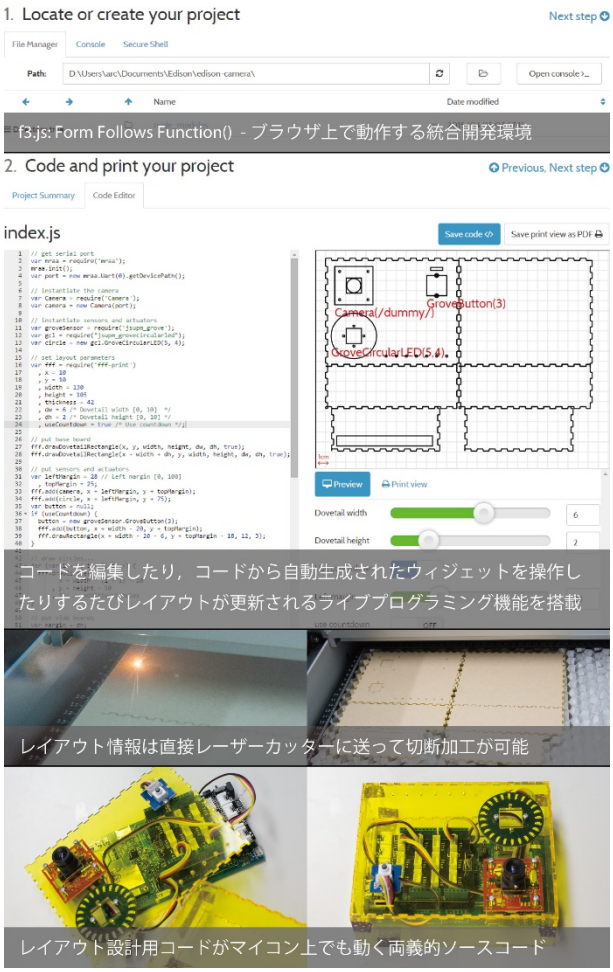


図 1.　f3.js を用いた開発プロセス（詳細は http://f3js.org）

マイコン(Intel Edison [1])に転送し，デバイスの振る舞いを制御できる．このようにブラウザとマイコンの両方で動作する**両義的ソースコード**は，各々のために用意された 2 種類の JavaScript インタプリタと，インタプリタの環境に応じて動作を変えるライブラリの組み合わせによって実現されている．

### 2.1 形状（レイアウト）のライブプログラミング

f3.js のプログラミングのための画面では，左側にソースコードが，右側にレーザーカッターで切り出したいレイアウトがプレビュー表示される．筐体に載せるセンサーやアクチュエーターは，`var Camera = require('jsupm_grovescam'); var c = new Camera();`のようにソースコード中でドライバをインスタンス化し，`fff.add(c, 100, 360);`のように f3.js が提供するライブラリの API に引数として渡すことで，レイアウト上に配置できる．レイアウトのプレビューには，レーザーカッターで実際に切断加工される線だけでなく，ユーザの利便性のために，筐体に載せるセンサーやアクチュエーターの名前や外形も重畳表示できる．これらの内容は，ブラウザ上で動作する JavaScript インタプリタによってソースコードを実際に実行することでレンダリングされている．

プレビューの内容はソースコードを変更するたびに更新されるため，ユーザは，どのような形状のデバイスを作成したいか逐一確認しながらソースコードを記述していくライブプログラミングを行える．また，ソースコード中の変数宣言後に `var useCountdown = false /* カウントダウン機能の有無 */;`のようにコメントを書き入れることで，プレビューの下に値の型に応じたスライダーやスイッチのようなウィジェットを表示できる．ウィジェットを操作するとレイアウトがインタラクティブに更新され，対応する変数の値も書き換わる．

プレビューの上にある印刷ボタンをクリックすると，現在のレイアウトが PDF ファイルに変換される．このファイルをレーザーカッターに送信すると，アクリル板などを設計したレイアウトで切り出せる．

### 2.2 機能（マイコンの振る舞い）のプログラミング

2.1 節で紹介したソースコードは，そのままマイコン上でも動作する．ただし，ブラウザ上ではレイアウトのプレビューが表示されるのに対し，マイコン上にはプレビューを表示する画面がない．その代わり，センサーやアクチュエーターのドライバのインスタンスは，実物を操作するためのメソッドやプロパティを持っており，これを呼び出すことでマイコンの振る舞いをプログラミングできる．例えば，カメラなら `c.capture(callback);`，ロータリーエンコーダーなら `re.getValue();`と書いて，写真を撮影したり角度の情報を取得したりできる．

マイコンの振る舞いをプログラミングするためのメソッドやプロパティはブラウザ上で読み込まれるライブラリでは未定義だが，ブラウザ上で動作するインタプリタは未定義のメソッドやプロパティは無視するように設計してあるため，レイアウトのプレビューが問題なく表示される仕組みになっている．

## 3 関連研究と議論

DressCode [2]はデザイナーが 2 次元形状を自由に描けるドメイン固有言語のための開発環境である．ソースコードを入力するとすぐ形状が見られるライブプログラミングを実現している．ShapeJS [3]は 3 次元形状をモデリング可能なライブラリだが，ブラウザ上で動作するプレビュー機能の付いた開発環境も提供している．本研究は形状の開発環境を機能の開発もできるように拡張する手法の提案と見なすことができ，これらの既存研究にも適用できる．

近年のマイコン用統合開発環境は本研究に限らず形状の開発機能を取り込もうとしており，例えば Autodesk 123D Circuits [4]は，Arduino ハードウェアといくつかの電子部品の振る舞いをエミュレートでき，簡単な基板設計も行える．.NET Gadgeteer [5]は Visual Studio 上でマイコンとその他モジュールの接続関係を表示できる他，市販の高級 CAD ソフトウェアと連携してレイアウトの制作を支援する機能を実装している．本研究の独自性は，CAD を用いず，両義的ソースコードで形状と機能を両方開発できる点にある．これにより，単一の JavaScript ファイルから IoT アプリケーションのハード・ソフトウェアを丸ごと全て出力できる特長がある．将来的には， GUI で一般的なレイアウトマネージャなどの仕組みを実装してエンドユーザによるレイアウトの直接操作を実現し，スライダーなどのウィジェットよりも容易なカスタマイズを支援する予定である．

## 参考文献


[1] Intel Edison. 2014. Retrieved 4/1/'15 from http://www.intel.com/content/www/us/en/do-it-yourself/edison.html

[2] Jacobs, J., and Buechley, L. Codeable Objects. In *Proc. of CHI'13*, 1589-1598.

[3] Shapeways. ShapeJS. http://shapejs.shapeways.com

[4] Autodesk 123D Circuits. 2015. Retrieved 4/1/'15 from http://www.123dapp.com/circuits

[5] Nicolas Villar, et al. .NET Gadgeteer. In *Proc. of Pervasive'12*, 216-233.