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

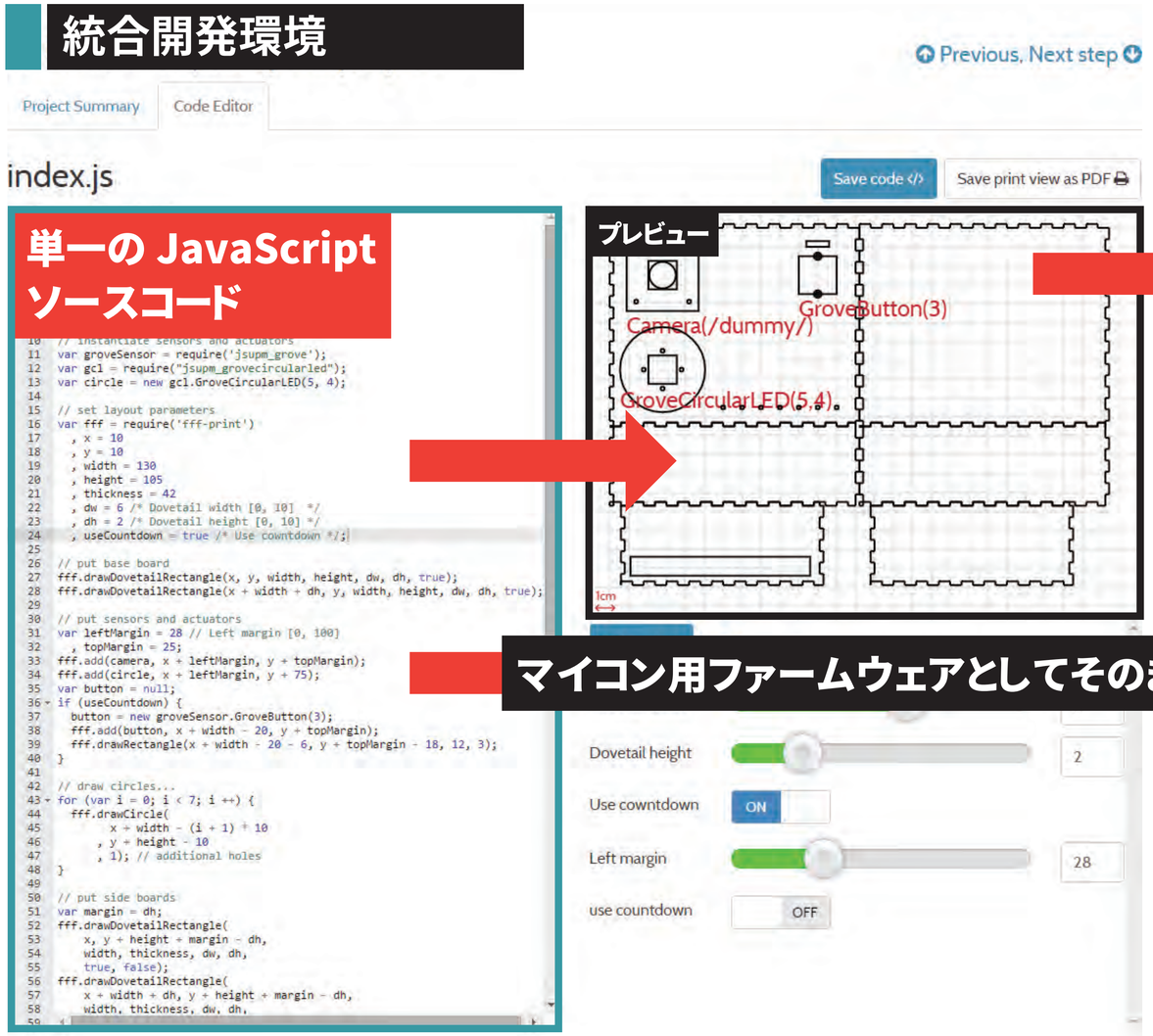
WISS' 15 (3-R07)

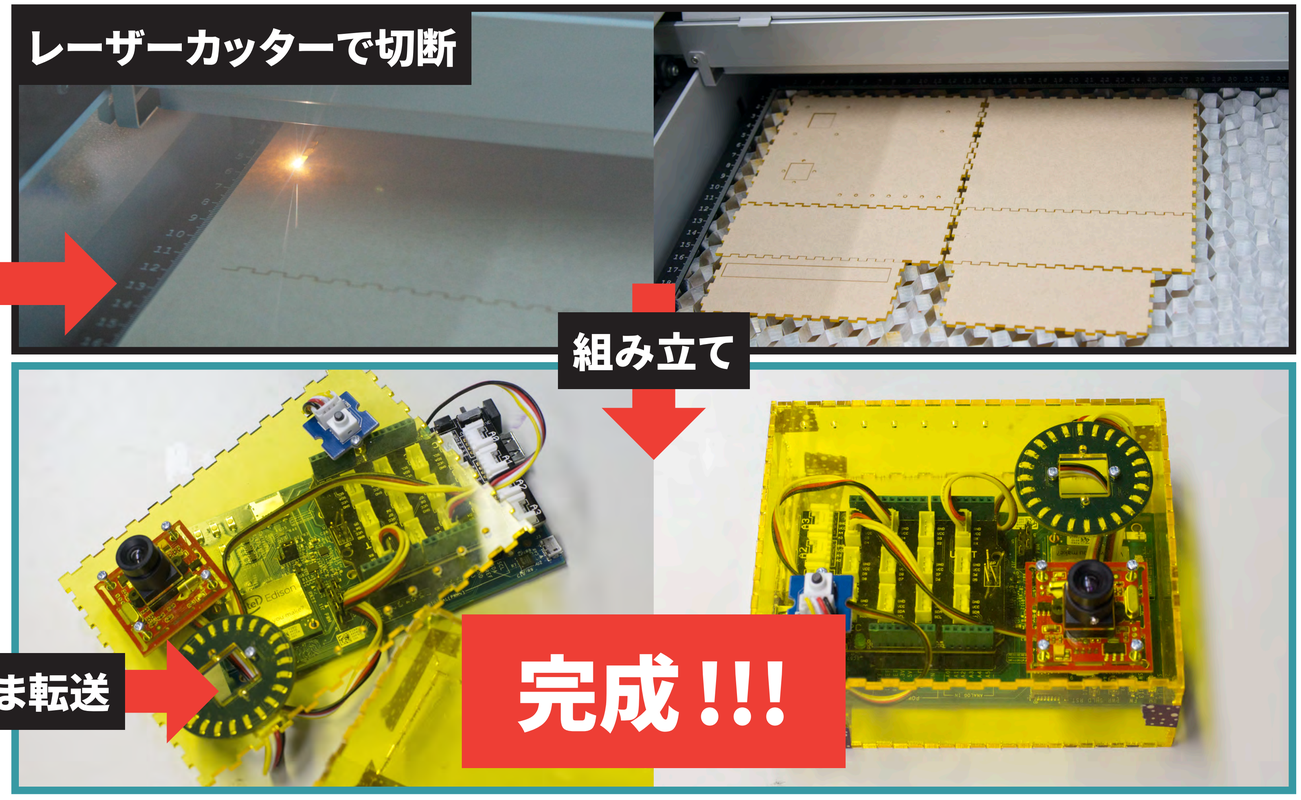
加藤 淳，後藤 真孝
産業技術総合研究所

研究のポイント

- JavaScriptを書けば、マイコンの振る舞いだけでなく筐体設計まで完成する**統合開発環境**
- GUIアプリ開発のように、筐体表面にセンサやアクチュエータを容易に配置できる**API**
- エンドユーザでも、用途に合わせて機能と形状をカスタマイズできる**GUIウィジェット**

統合開発環境





既存手法との比較

- ソースコードと配線可視化の双方向編集 (.NET Gadgeteer)
- GUI で配線・振る舞いシミュレーション (Autodesk 123D Circuit)

⇔ 配線設計に留まらず エンドユーザが触るUI 設計まで支援

- 配線を手描きしてモジュールを配置できる開発環境 (PaperPulse)
- センサやアクチュエータから筐体を自動生成できる CAD (Enclosed)

⇔ プログラミングを用いることで 詳細に再利用性の高い設計が可能

API・GUI ウィジェット

- センサやアクチュエータのネジ穴などのレイアウト情報を提供（ユーザが拡張可能）

```
var button = new gs.GroveButton(2); // もともと button.getValue() などが使える
fff.add(button, 50, 40); // これで座標 (50, 40) にボタンを配置できるようネジ穴が開く
```

- 立体形状を定義すれば、複数の切断面を、繋ぎ目のギザギザも含め自動的に生成

```
var rect = fff.drawRectangle(5, 5, 60, 20); var planes = rect.extrude(50);
// 矩形を 50mm 押し出した立体形状を作り、6 面を生成
```

- 変数にコメントをつけると、機能と筐体をカスタマイズできるGUIウィジェットが出現

```
var useCountdown = true; // カウントダウン機能を使う
if (useCountdown) { /* 変数値次第で機能と筐体レイアウト両方を差し替え可能 */ }
```

http://f3js.org で一般公開予定！（デモ動画など情報配信中）

仕様

※灰色は今後実装予定・検討中の機能

対応マイコン : Intel Edison, Raspberry Pi, Tessel.io, Arduino Yun
対応モジュール : Grove システム , ユーザ向け GitHub リポジトリ公開予定
OS: Windows 7/8/8.1/10, Mac OS X, Linux