

# 拡張②…JavaScriptでマイコン制御用ブロックを自作する

そーたメイ

## ● Lチカ・ブロックを追加する拡張機能を作る

第2部 第2章の図1と同じプログラムを拡張機能 test.js として作成します。プログラミング言語は JavaScript です。ソースコードをリスト1に示します。作成したファイルは、適当な場所に保存しておきます。サンプル・コードは次の URL から入手できます。

<https://www.cqpub.co.jp/interface/download/contents2022.htm>

つくるっちを Google Chrome で開きます。拡張機能から [uno.ESP32] (micro:bit の場合は [micro:bit]) と [外部拡張読み込み] を選択し、リスト1の通り作成した test.js を選択して [load] をクリックします。成功すると、図1のブロックが表示されます。ブ

ロックをクリックすると、LEDが2回点滅します。

読み込みや実行が失敗するときは、[F12] キーを押して表示される JavaScript のログを確認してみましょう。拡張機能を再読み込みするときは、[F5] キーを押してウェブ・ページを更新してください。

## ● 拡張機能 JavaScript (test.js) の内容

### ▶ (1) ブロック定義

リスト1の①が図1のブロックの定義です。今回のブロックでは、[ポート] を引数 ARG1 として関数 (opcode) blinkLED を呼び出します。各パラメータの定義は表1の通りです。

リスト1 Lチカ・ブロックを追加する拡張機能を記述した JavaScript コード (test.js)  
拡張機能として読み込めば、第2部 第1章の図1と同じ実行内容のLチカ・ブロックが追加される

```
var extName = 'test'; // デバイス拡張名 _ や、などの記号禁止

var ext = class {
  constructor(runtime) {
    this.runtime = runtime;
  }

  getInfo () {
    return {
      id: extName,
      name: extName,
      blocks: [ // ブロック定義
        {
          blockType: BlockType.COMMAND, opcode: 'blinkLED', text: 'ポート[ARG1]',
          arguments: {
            ARG1: { type: ArgumentType.NUMBER, defaultValue: 13 },
          }
        },
      ],
      menus: [], // メニュー定義 (未定義);
    };
  }

  blinkLED(args) {
    let port = args.ARG1*1; // ブロックのARG1が取得できます
    const this = this;
    return this.runtime.dev.comlib.digiWrite([
      {port:port, level:1} // digitalWriteと等価
    ])
    .then(() => new Promise(resolve => setTimeout(resolve, 1000))) // delayと等価
    .then(() => _this.runtime.dev.comlib.digiWrite([
      {port:port, level:0}
    ]))
    .then(() => new Promise(resolve => setTimeout(resolve, 1000)))
    .then(() => _this.runtime.dev.comlib.digiWrite([
      {port:port, level:1}
    ]))
    .then(() => new Promise(resolve => setTimeout(resolve, 1000)))
    .then(() => _this.runtime.dev.comlib.digiWrite([
      {port:port, level:0}
    ]));
  }
}
```