2021年 お勧め開発環境
VS Code

第6回 自作拡張機能のデバッグと操作性カスタマイズ

加藤 史也

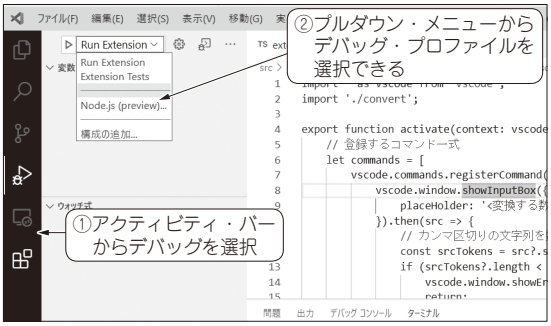

図1 単体テスト用のプロファイルが最初から用意されている

Visual Studio Code（以下VS Code）は，ソースコード・エディタの他に，ソフトウェア開発に必要な機能を一通り取りそろえた統合開発環境です．

言語やターゲット・デバイスに応じてさまざまな拡張機能を組み合わせることで，PCやマイコンのプログラミングを快適にできます．

前回は，独自の拡張機能を作る方法を紹介しました．今回は，作った拡張機能をデバッグする方法や，さらに使いやすくカスタマイズする方法を紹介します．

完成した機能を，誰でも使えるように公開する仕組み（Visual Studio Marketplace）も用意されているので，その方法も紹介します．

拡張機能をデバッグする

● デバッグ用メッセージはコンソールに表示される

独自の拡張機能は，ひな型を基に作ります（詳細は2021年9月号を参照）．ひな型の時点で既に，ソースコードの中にコンソールへのログ出力の埋め込みがありました．これは拡張機能の開発を行っているVS Codeのデバッグ・コンソールに表示されます[注1]．作っている機能が，VS Codeから呼び出されているタイミングが不明瞭だったり，特定の変数を表示させつつデバッグしたりしたい場合は積極的に埋め込むとよいです．

内容を表示させつつデバッグできます，これ以上に細かいことをしたい場合は後述のデバッガの利用を推奨します．

● ブレーク・ポイントや変数内容の表示

拡張機能開発でもデバッガを利用できます．ソースコード上の停止させたい行でF9キーを押してブレーク・ポイントを設定します．ブレーク・ポイントが設定された行の横には，赤い丸印が表示されます．この状態でデバッグを開始すると，ブレーク・ポイントの行で処理を行おうとした際に停止し，1行ずつ実行させたり現在の変数の内容を確認したりできます．

● 拡張機能開発ホスト上で単体テストが実行できる

ひな型作成ツールのYeomanで生成したひな型には，拡張機能開発ホスト上で単体テストを実行するテスト・スイートの実装も含まれています．実行するには**図1**のようにデバッグ・プロファイルをRun ExtensionからExtension Testsに切り替えて実行するだけです．

テストを一通り実行した後，自動的に拡張機能開発ホストのVS Codeが閉じられ，**図2**のように結果が表示されます．

ひな型として生成した状態では，
`src/test/suite/**.test.ts`
を一通り実行するように実装されているため，自分でテストを追加するには既存の，
`src/test/suite/extension.test.ts`
を編集するか，`src/test/suite/`に命名規則を守ったファイルを作成して実装します．今回は，
`src/test/suite/convert.test.ts`
を実装して検証しています．

注1：拡張機能開発ときは，開発用と動作確認用に2つのVS Codeを実行させる．