

組込み開発環境のグローバル・スタンダード

IARの統合開発環境

開発環境の標準化で
製品開発の効率と信頼性を高める

これまで日本市場では独自マイコンが広く利用されていたこともあり、さまざまなコンパイラやデバッガが利用されていました。近年、急速に普及しているARMマイコンへの移行やその他の異なるマイコン間での移行時に、コンパイラやデバッガの変更という課題が発生しています。この変更により操作性の違いやオプションの違い、これまで動作してきたソフトウェアの確認など開発者の負担となる作業が発生します。ここでは、世界で最も幅広く汎用マイコンをサポートしているIAR Embedded Workbench (以後EW) を使うことで、こうした負担を減らし、より効率的で信頼性の高いソフト開発を行うためのいくつかの機能を紹介します。

まず、コンパイラやデバッガの変更時には、操作性や機能が変化することにより、プロジェクト立ち上げ時の調査や習得が必要になりますが、EWではさまざまなマイコンで同じGUIや機能を実現しており、エディタやコンパイルに関する機能はすべてのマイコンで基本的に共通ですので再調査や再習得が不要になります。コンパイラ自体も同じ思想で作られているためマイコン毎に差異が少なくなっています。たとえば、charがunsignedで統一されていることなどが移植性を向上するポイントになっています。また、charについてはオプションによりsignedに変更も可能という点も基本的にすべてのマイコンに共通です。

次に、EWのC-SPYデバッガは、C-SPYマクロを使って機能を拡張することができます。

たとえば、デバッグ時にリセットをかけたときは、通常はソフトウェアリセットになりますが、リセットピン経由でリセットをするためのマクロを記述することができます。

C-SPYマクロを使って“やりたいこと”を資産化することで、ユーザー独自のC-SPYデバッガを作り、ソフトウェアのデバッグを効率的に実施することができます。

C-SPYマクロでは、ファイルにデータを書き出すことができますし、ファイルからデータを読み込むことができますので、テストをする際にあるメモリ領域を書き換えたり、または、解析用にある領域のメモリ値をファイルに書き出すこともできます。

EWではC言語で規定されたassert文をサポートしており実行時に違反があった場合には、メッセージを表示します。ただし、assert文を有効にした場合と無効にした場合で生成されるコードに差がでるため、厳しい検査レベルを要求される場合には利用できないケースがあります。

デバッグ手法のマクロ化で
機能拡張と手順の共通化を実現

そうした場合にも、C-SPYマクロを使って、実行中の任意の場所でチェックを実現することができます。また、単純にC-SPYマクロを実行してログで結果を確認するだけではなく、違反が発生したときに動作を停止し、解析することも可能です。

更に特筆すべきは、C-SPYマクロを2階層で作成することで、異なるマイコンでも利用可能となる点です。

関数の入り口で引数をチェックするアサーションをC-SPYで書くことを考えると、引数はどうしてもマイコンにより異なるため、C-SPYマクロで書いたチェックの再利用ができません。

しかし、マイコン依存部を分離することで、異なったマイコンに対して再利用が容易になります。こうした機能を使うことで人手によってチェックする作業を減らしつつ、デバッグ手法の共通化を実現することができます。

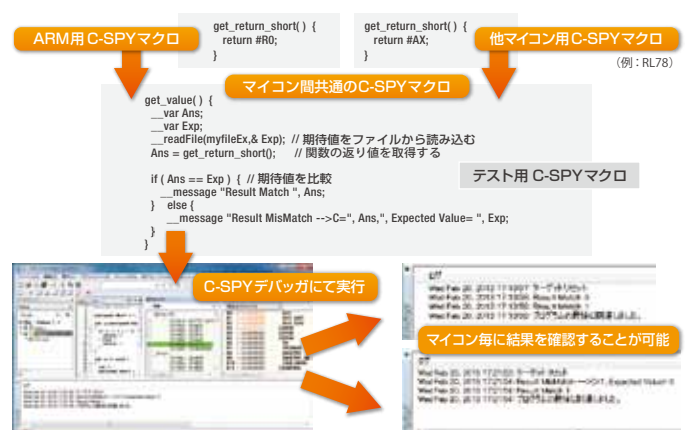
このような機能は当然コンパイラによって異なるため、使用するコンパイラが変わる前提ではなかなか積極的に使わないようにする傾向があります。しかし、EWを標準コンパイラとすることで、ユーザーも積極的にC-SPYマクロを組むことができ、資産として運用することが可能となります。

お問い合わせ先

IARシステムズ株式会社

〒101-0041 東京都千代田区神田須田町1-21-5 C-5ビル5階

TEL : 03-5298-4800 FAX : 03-5298-4801

Web : <http://www.iar.com/jp/> Web フォーム : http://www.iar.com/contact_jp

図：CSPYマクロを用いたマイコン間の共通テスト環境構築例