

マイコンソフト開発ツール間連携インターフェイス Tool Link

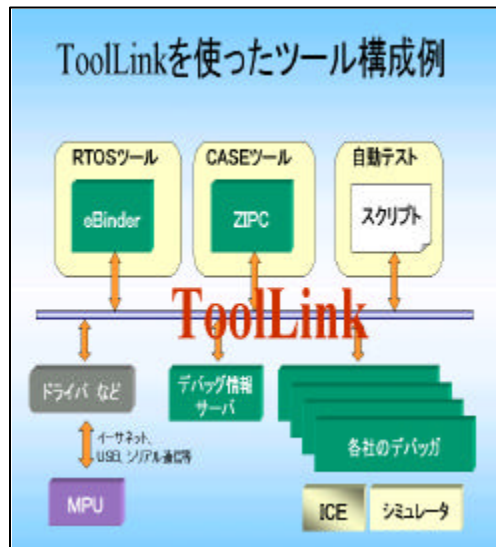
松下電器産業株式会社 半導体社 開発本部 マイコン開発センター
第四開発グループ 第三開発チーム 主任技師

松木 敏夫

1. マイコンソフト開発環境の グレードアップ

パソコンという共通プラットフォームが使えるパソコンアプリケーションの開発環境と比べると、マイコンソフトの開発環境はどうしても見劣りしてしまう。カスタム性が強いマイコンシステム開発ではツールコストが上がる傾向があるために、多くのエンジニアに必要十分なツールが行き渡っていないのが現状ではないかと思う。しかし、高度化するテクノロジーへの対応と製品短期開発、差別化、そして（果てしない？）コストダウンの追及が私たちエンジニアの任務である。全てのマイコンシステムエンジニアはツールコストの増大は抑えつつ、より良い開発環境を常に探っていくことが必要だと考える。

製品開発において開発効率を上げるための正攻法とも言うべき手段は、共通化であり、部品化である。マイコンソフト開発環境においても同様で、共通化や部品化により無駄を省けるはずである。たとえばデバッガを例にとると、一つのデバッガでどのようなマイコンでもデバッグできれば十分ではないだろうか。もちろ



んユーザ毎に好みのデバッガはあるであろうから、デバッガは何種類存在しても良いが、1ユーザの立場からすればデバッガは一つで十分である。つまり、一つのデバッガ GUI で様々な ICE を制御するのが良い。すなわち、デバッガや ICE を部品のように扱い、自在に組み合わせでゆけるようにするのである。CASE ツールも同じように扱うことはできないだろうか。いろいろな手法の CASE ツールをうまく組み合わせでゆければ、開発しようとするマイコンシステムに最適な開発環境を得ることが出来るに違いない。

さて、本稿でこれから説明する ToolLink は、ツール間の接続を行うための通信機構とプロトコル仕様を無償で提供するものである。現時点ではデバッガや CASE ツールなどの接続を対象としたものとなっているが、ToolLink 通信機構は拡張性が大きく、デバッガや CASE ツール以外のツール間連携にも利用できる。

理想のマイコンソフト開発環境の構築にきっとこの ToolLink が役立ってくれるものと考えている。

2. ツール間の接続方法

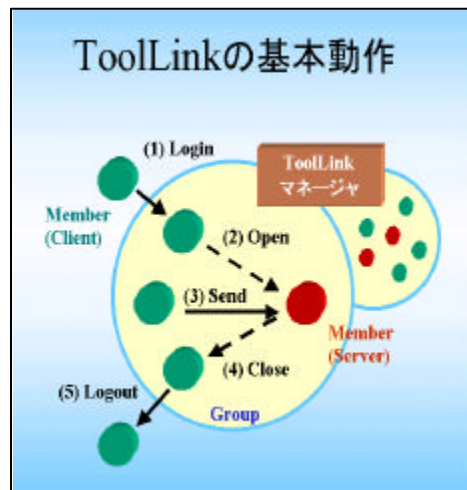
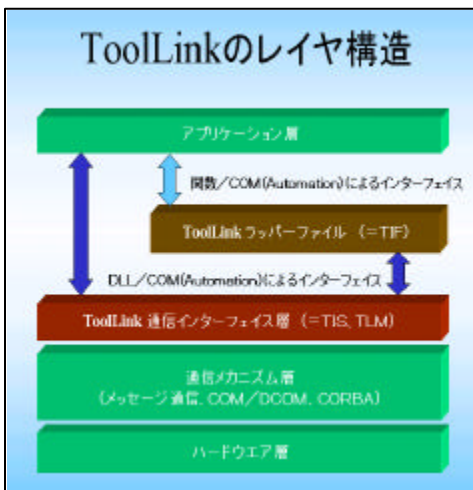
現在提供可能な ToolLink 通信機構ではパソコン内のアプリケーション間の通信（インプロセス通信とアウトプロセス通信）およびパソコン間のアプリケーション間通信（リモート通信）ができるようになっている。ToolLink で通信機構をつかさどるのは ToolLink Manager（TLM）と呼ばれるアプリケーションと ToolInterfaceSocket(TIS)と呼ばれるダ

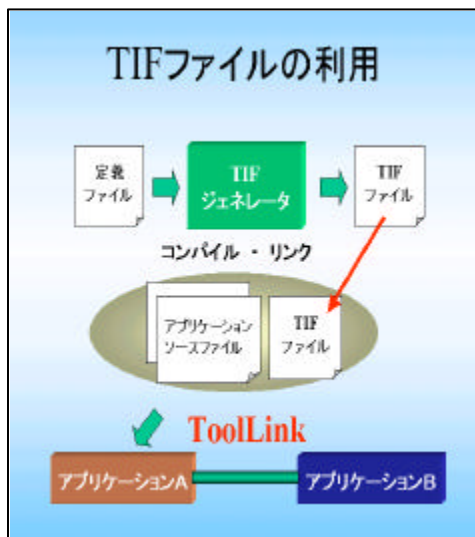
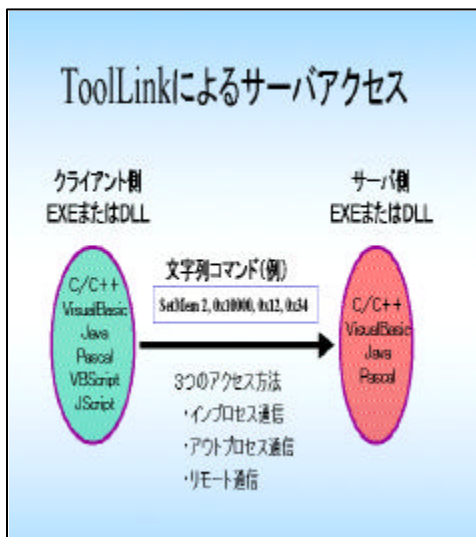
イナミックリンクライブラリ（DLL）である。これらは Windows に標準装備されている COM という通信機構を使ってツール間の通信を実現しているが、TIS や TLM によって内部の通信メカニズムを隠蔽しているために、ツール開発者やツールユーザはリモート通信の時以外は COM を意識する必要がない。

ツールは次の手順でツール間接続を行うことができる。

- (1) ログイン（ToolLink への参加）
- (2) オープン（サーバとなるツールとの接続を行う）
- (3) 送信（コマンドの送信）
- (4) クローズ（サーバとの接続を閉じる）
- (5) ログアウト（ToolLink から抜ける）

ToolLink によるツール間のやりとりは非常に簡単で、使用する言語にもよるが基本的には5つの関数を覚えておけばコマンドを送ることができる。





ツールは最初に自分の名前を付けて(1)ログインする。クライアント(サービスを受ける側)となるツールはサーバ(コマンドを受けて応答する側)のツールの名前を(2)オープンの際にパラメータとして指定する。(3)送信ではオープンしたサーバに対してコマンドを送り、サービスを受けるのが基本だが、不特定多数のサーバに対して一斉にメッセージを送ることもできる。コマンド送信では同期式送信及び非同期式送信の利用が可能である。

WindowsScriptingHost (WSH) を使った VBScript によるクライアントの例を示す。WSH を使うとスクリプトファイルだけでサーバを操作できる。

この例では、"RTOSAnalyzer"というツールが"DebuggerA"というサーバを呼び出して接続し、クライアントがサーバに対して "SetMem 2, 0x10000, 0x12, 0x34" というメモリ書き込みコマンドを送っている。このように ToolLink ではコマンドを文字列でやりとりしている点に特徴がある。

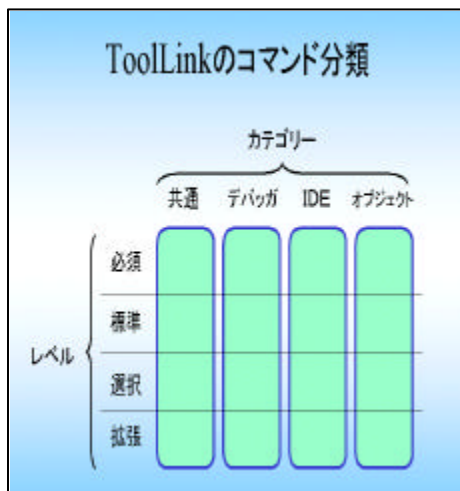
VBScriptを用いた例 (Sample.vbs ファイルの内容)

```
Dim server          ' 変数の宣言
set server = WScript.CreateObject("ToolLinkManager.TScript") ' TLMの呼び出し
server.Login "RTOSAnalyzer" ' ログイン
server.Open "DebuggerA"    ' オープン
server.Send "SetMem 2, 0x10000, 0x12, 0x34", 0 ' コマンド送信
                        ' 0x10000番地に0x12と0x34のデータを2バイトアクセスで設定する
server.Close          ' クローズは記述してもしなくても良い。
server.Logout        ' ログアウトも記述してもしなくても良い。
```

ここではVBScriptの例を紹介したが、C言語やC++、VisualBasicも基本的な操作は同じであり、コーディングは簡単である。文字列を使ったやり取りはバイナリデータを使った送受信に比べるとややオーバーヘッドが大きくなるが、コマンド追加の容易性やコマンドフォーマットの柔軟性、可読性のメリットは大きい。これはインターネットの世界でHTMLやXMLによって実証されている事実である。しかし、文字列を数値に変換するなどの処理をコーディングするのはコマンド数が多くなると結構面倒なため、ToolLinkでは文字列と数値を相互変換処理を行うプログラムソースファイルを生成するツール(TIFジェネレータ)も提供している。このツールが出力したファイルをアプリケーションにリンクすれば文字列を直接操作する処理をコーディングする手間が不要になるので便利である。

3 . ToolLink コマンド仕様

共通の通信手段があるだけではツールの接続は楽にはならない。その通信手段を使って何をやりとりするかも非常に重要である。ToolLinkではコマンド仕様を定めており、サーバとなるツールはこのコマンドを受けて応答することができる。ツールの種類によって必要なコマンドは異なるが、現在は4つの種類のサーバを想定した4つのコマンドカテゴリーを定義している。それぞれのコマンドカテゴリーの中でもさらに最もプリミティブな



「必須コマンド」、シンボルが扱える「標準コマンド」、「選択可能コマンド」、「拡張コマンド」にコマンドに分けて順位づけがされている。ツール開発者は、最初は必須コマンドから実装し、ツール間連携の具合を確かめてから、次第により高度な連携を実現するのが良いと思われる。ただし、サーバ側にコマンド実装の自由度を持たせているために、不特定のサーバと接続するようなクライアントツールはサーバがどのようなコマンドを実装しているかを実行時に確認する必要がある。現在定義しているコマンドカテゴリーは次のようなものである。

(1) 共通コマンドカテゴリー

全てのサーバで実装すべきコマンドであり、現在はアプリケーションの終了コマンドのみを規定している。

(2) DBG(デバッグ)コマンドカテゴリー

デバッグ(サーバ側)で実装しなければならないコマンド。ICEデバッグな

どでは ICE が実装している機能によってサービスできるコマンドが異なる。

主なコマンド：メモリ操作、レジスタ操作、実行制御、ブレークポイント制御、アクセスブレーク制御、シンボルテーブル参照、デバッグ情報の参照、実行ファイルのロード、関数実行、トレース制御、I/O操作、デバッグ画面の制御

(3) IDE (統合開発環境) コマンド

カテゴリー

IDE(サーバ側。エディタやビルドツールを含んでいるもの。)で実装すべきコマンドがある。

主なコマンド：プロジェクトファイルの制御、エディタの制御、ビルドの制御、デバッグモードの制御

(4) オブジェクト操作コマンド

カテゴリー

シミュレータのようなツールの上で動作するオブジェクトを操作するためのコマンドである。

(5) DBG (デバッグ) 拡張コマンド

カテゴリー

DBG コマンドクラスを拡張したもの。

ToolLink ではツールの実装工数が大きくならないようにコマンドを厳選してある。サーバ、クライアントいずれでも結構なので一度試していただきたい。

4 .TISC 委員会

ToolLink の仕様策定及び普及活動は TISC 委員会というものを設置して日本電気(株)、キャッツ(株)、エルグ(株)、松下電器産業(株)のツール開発者が運営を行っており、また、国内半導体メーカーのツール開発担当者やツールベンダー様、ツールのユーザ様にも参集していただいて仕様策定作業などを行っている。ToolLink の問い合わせ先は下記のとおり。

松下電器産業(株) 松木敏夫

matsuki@mpd.csdd.mei.co.jp

ToolLink メールングリスト

tisc-all@mctcs5.ucom.lsi.nec.co.jp

