

ソフトウェアテストシンポジウム 2010 東京

Engineered IT Solutions

Windows Embedded CEに見る、組込み開発テスト環境

平成22年1月29日

ソリューション技術本部
組込制御ソリューション統括部 組込第1技術部

中田 佳孝



安川情報システム株式会社



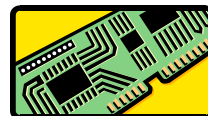
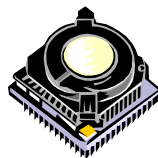
- ◆ 組込み開発の現状
- ◆ Windows Embedded CE概要
- ◆ 組込み開発でのテスト懸念点
- ◆ Windows Embedded CEに見るテスト環境
- ◆ テストツール使用時の注意点
- ◆ まとめ

Windowsは米国Microsoft Corporationの米国及びその他の国における登録商標または商標です。
本資料に記載されている会社名及び製品名は、各社の登録商標または商標です。

組込システムに対する市場の性能ニーズは高い ⇒ ハイスペックOSの搭載

◆ ハードウェアの高機能化

- CPUクロック、メモリサイズ ⇒ 大
- 操作性や見栄えのいいGUI



◆ デバイスの多様化

- USB, IEEE1394 etc...



◆ サービス・プロトコルの多様化

- Webサーバ、FTPサーバ、SNMP etc...

⇒ ハイスペックOS搭載により

スクラッチでの開発ボリュームを低減

Windows Embedded CEにより、
容易に高機能・高性能なシステム構築が可能。

- ◆ 32bit CPU (ARM, MIPS, x86, SH4) 用の **リアルタイムOS**
 - ◆ カーネルは **PC版Windowsとは別設計**
 - ※システムコールはWin32APIのサブセット
 - ◆ 約600個のコンポーネントが標準で付属
 - 対応デバイス : シリアル, NIC, IrDA, Bluetooth, 無線 LAN etc...
 - 対応プロトコル : TCP/IP & v6, VoIP, VPN etc...
 - 各種サーバ : Web, FTP, SMB etc...
- ⇒プロダクトライン開発に有利**
- ◆ 統合開発環境 (**Visual Studio 2005 + アドイン**) での開発
 - ◆ 製品サポート: **10年**、製品供給: **15年**

ARMはアーム・リミテッド社、MIPSはMIPS社、x86はIntel社、SHはルネサステクノロジー社の登録商標です。
BluetoothはBluetooth SIGの登録商標です。

質を確保するためにも
効率よいテストができる仕組みが求められている。

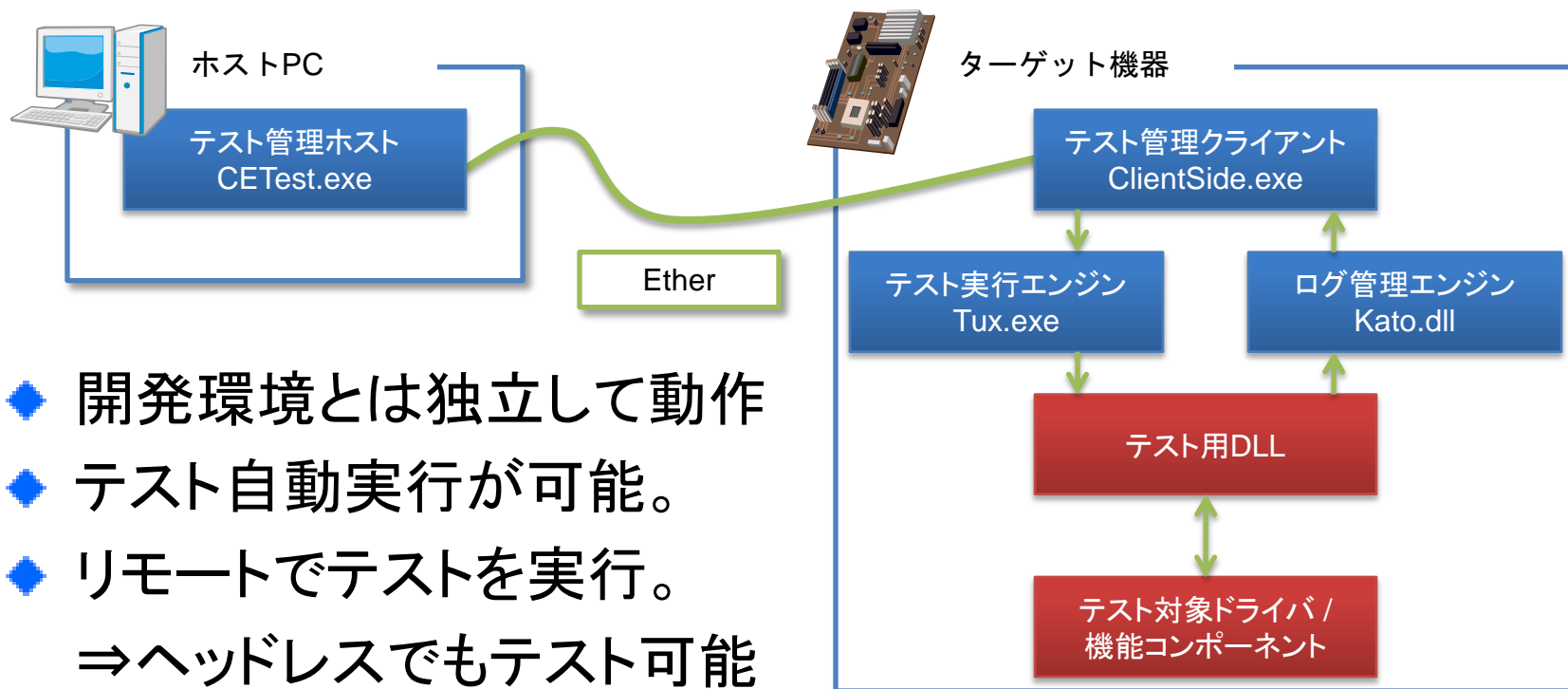
ハイスペックOSは搭載されるようになってきたが...

- ◆ ステップ数増加＝検証ボリュームの増加
- ◆ (従来より多くはなったが) 限定されたリソース
- ◆ 長時間運転
- ◆ 稼働中の電源OFF ⇔ ON
- ◆ データ保護
- ◆ 機器と連動 ⇒ 高安全性・高品質

- ◆ デバイスドライバ用テストツール
 - Windows CE テストキット (CETK)

- ◆ アプリケーション用テストツール
 - Application Verifier (AppVerifier)

デバイスドライバや各種コンポーネント用の Testingフレームワークを提供！

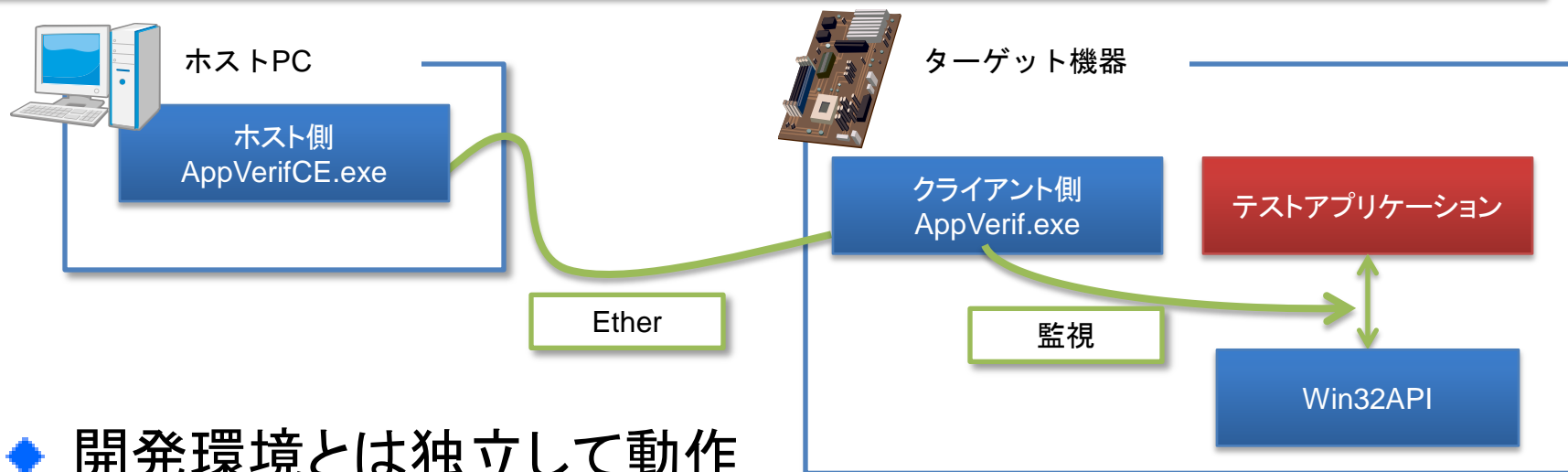


- ◆ 開発環境とは独立して動作
- ◆ テスト自動実行が可能。
- ◆ リモートでテストを実行。
⇒ヘッドレスでもテスト可能
- ◆ 標準的なデバイスに対するテストルーチンがあらかじめ付属
- ◆ 独自のテストスイートを追加・実行・再利用可能。

Test Cases for the Waveform Audio Driver Test

Test case	Description	Test case	Description
100	Build Verification Test	3001	Capture
1000	Easy Playback	3002	Capture Notifications
2000	Playback Capabilities	3003	Capture Buffer Freed During Capture
2001	Playback	3006	Capture Initial Latency
2002	Playback Notifications	3007	Capture Initial Latencies Series
2003	Playback Using Extended Functions	3008	Capture Multiple Streams
2004	Playback Buffer Freed During Playback	6000	Playback Mixing
2005	Playback Sample Rate Reporting	6001	Capture Mixing
2006	Playback Initial Latency	7000	International Audio File Names
2007	Playback Initial Latencies Series	8000	Verify Handle Validation
2008	Playback Interoperability	8001	Verify Device ID Validation
3000	Capture Capabilities	8002	Verify GUID Validation

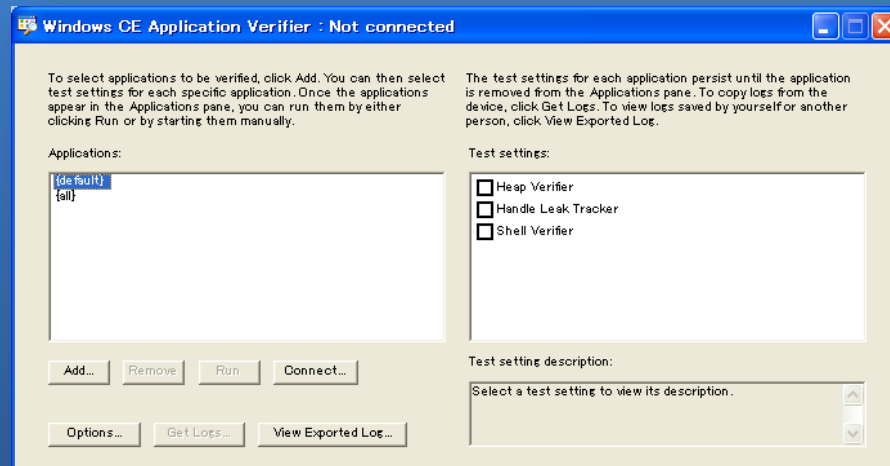
アプリケーションが正しくターゲット資源を使用しているか
効率的にチェックできる。



- ◆ 開発環境とは独立して動作
- ◆ 関数フックによるAPI監視 ⇒ アプリ側の変更不要
- ◆ 監視結果(例外情報やCall Stack)をXMLでデータ出力可能
- ◆ スタンドアロン／リモートどちらでも実行可能。
(リモートで結果データの取得が可能)
- ◆ アプリケーションが終了できることが条件

ホスト側:

c:\Program Files\Microsoft Platform Builder\6.00
¥cepb¥wctk¥ddtk¥desktop¥AppVerifCE.exe



AppVerifierの起動

テスト対象アプリの
起動

テストシナリオ実施

テスト対象アプリ
クローズ(終了)

AppVerifierのログ
確認

- ・メモリーリーク及びヒープ領域破壊の監視
- ・同期オブジェクト等の各種ハンドルオブジェクトのリーク監視
- ・GDI及びユーザ定義オブジェクトのリーク監視

どんな仕組みで何が検証できるのか、
テストツールの動作もしっかり把握が必要。

- ◆ テストツールを使用する = 本番環境ではない。
 - CETKやAppVerifierがバックグラウンドで動作するため、CPUの使用率、メモリ/スタック使用量、通信負荷が少なからず増加。
- ◆ テストシナリオ内でどう使えるか？
 - 確認したい現象と、テストツールで確認できることはマッチするか？
⇒テストツールの仕組みの理解

◆ 組み込み開発の現状

- CPU, デバイスの高性能・多機能化 ⇒ ハイスペックOSの使用

◆ Windows Embedded CE事例

- アプリ、ドライバ向けに標準でテストツールが付属
- 開発環境とは独立して動作
- テスト用の仕組みをスクラッチで開発する必要がなくなる。
- テストスイートが再利用可能になる。
⇒効率よいテストが可能。
⇒QCDの確保につながる。

◆ テストツールの注意点

- テストツールを使う = 本番環境ではない
- ツールの仕組みを把握 ⇒ テストシナリオにマッチするか？



YASKAWA
INFORMATION

<http://www.yasknet.co.jp>