

設備機器システムのテストにおける CPU エミュレータの活用

Utilization of CPU Emulator in Software Test of Equipment System

黒岩 丈瑠†
Takeru Kuroiwa

久代 紀之‡
Noriyuki Kushiro

1. まえがき

近年、競争的に事業を推進するため、グローバルかつスピーディーな製品展開が求められることが多い。組込みシステムの開発においても、このような要求に対応するため、ソフトウェアプロダクトライン開発体制の採用・OSS の活用といった開発効率化策が積極的に進められている。

一方で、仮想化技術の進展、パソコンの CPU 性能の向上にともない、CPU エミュレータを実用的な速度で動作させることが可能となっている。

これらの現状を受け、QEMU[1]などの CPU エミュレータを活用し効率的な組込みソフトウェアのテストを実現する、実機レステスト環境が構築されつつある[2]。

しかし、実機レステスト環境を実際の製品開発に適用するには、製品の信頼性確保、開発現場へのスムーズな導入といった課題を解決する必要がある。

本論文では、空調・照明を初めとする機器の制御を行う組込みシステム（以下、設備機器システムと称す）におけるこれらの課題、及び CPU エミュレータを活用したシステムテスト環境による課題解決について述べる。

2. 組込みシステム開発における CPU エミュレータの活用

CPU エミュレータは、組込みソフトウェアをパソコン上で動作させることを可能にする。さらに、周辺デバイス（タイマ、メモリ等）のエミュレータを組合せれば、パソコン上で組込みボードを模擬することができる。

これにより、実機を使わずに組込みソフトウェアのテスト環境を構築（実機レス化）できる（図 1）。

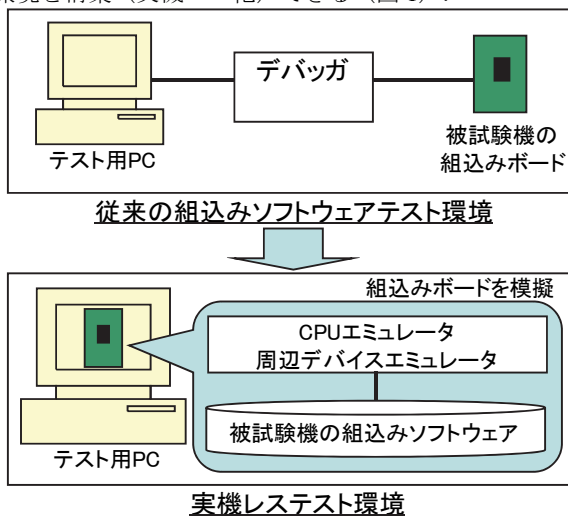


図 1 組込みソフトウェアテスト環境の実機レス化

この実機レステスト環境により、以下に示す効果が得られ、組込みシステムのテスト作業の効率化が期待できる。

- 組込みボードが完成する前に組込みソフトウェアのテストを開始できるため、開発期間の短縮が可能
- 環境の構築・送付が容易のため、大規模開発体制への対応が容易
- 組込みボードの故障の模擬ができるため、異常系に対するテストの実施が容易

3. 設備機器システム開発への適用における課題

設備機器システムにおいては、2 章に記載した実機レステスト環境のようなテスト効率化策を、システムテスト（ここでは、機器単体でのテストが完了した後に実施する、機器の動作の組合せにより実現される機能のテスト、を指す）に適用できると、特に効果が大きい。

何故ならば、以下の理由により、設備機器システムのシステムテスト項目数は非常に多くなるからである。

- システムテストの項目数は、機器の組合せパターンに応じて増加する
- 設備機器システムは、システムを構成する機器の機種、台数を顧客要求に合わせて調整できるようにしており、機器の組合せパターンが非常に多い

例えば空調システムでは、システム全体を管理する管理装置、空調機能を実現する室外機、室内機、換気ユニット、各ユニットを操作するリモコン等がシステム上に存在する（図 2）。仮に、これら 5 種類の機器がそれぞれ最大 10 台接続可能とすると、システムの組合せパターンは 10^5 通りにもなる。



図 2 設備機器システム（空調）のイメージ

しかし、実機レステスト環境をシステムテストに適用するには、以下の課題がある。

- 製品の信頼性に関する課題
設備機器システムには生活の基盤を担うものが多く、非常に高い信頼性が要求される。このためシステムテストでは、通信・操作などソフトウェア外の要因により非同期に発生するイベントについても、リアルタイム性・並行性・異常対応状況などを確認する。よって実機レステスト環境でも、これらについて確認できるようにしておく必要がある。
- 開発現場への導入の容易性に関する課題
設備機器システムのシステムテストでは、ユーザ視点でシステムが正常に動いているか、を確認する。

† (株) 三菱電機, Mitsubishi Electric

‡ 九州工業大学, Kyusyu Institute of Technology

よって実機レステスト環境をシステムテストの現場にスムーズに導入するためには、各エミュレータの入出力値が確認できるだけでは不十分で、実機と同等のインターフェースを供給し、従来のテスト環境と同じようにシステムの操作・状態確認をできるようにする必要がある。

そこで筆者らは、上記の課題を解決し、設備機器システムのシステムテストを効率化する、CPU エミュレータを活用したシステムテスト環境を構築している。以下、そのテスト環境の開発及び評価について述べる。

4. CPU エミュレータを活用した設備機器システムのシステムテスト環境

4.1 概要

設備機器システムにおける従来のシステムテスト環境を図 3 に、筆者らが開発しているシステムテスト環境を図 4 に示す。

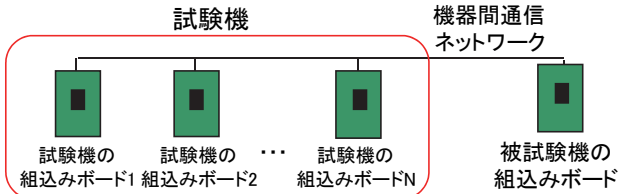


図 3 従来のシステムテスト環境

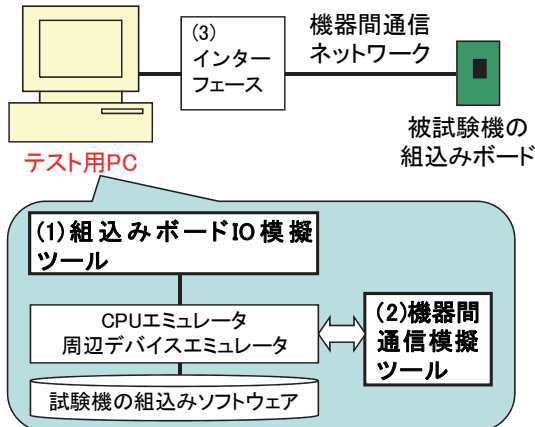


図 4 開発しているシステムテスト環境

開発しているシステムテスト環境は、システムテストにおける試験機を実機レス化するものである。構成要素は以下の 3 点である。

- (1) 組み込みボード IO 模擬ツール
組み込みボード上の各種入出力インターフェースを、PC 画面上に模擬するツール
- (2) 機器間通信模擬ツール
設備機器システムの機器間通信を、実機と CPU エミュレータとの間で実施可能にするツール
- (3) インターフェース
パソコン用通信のプロトコル (ETHERNET¹, USB 等) と、設備機器システムの機器間通信プロトコルとの、物理層変換を実施する機器

¹ ETHERNET は、富士ゼロックス (株) の登録商標である。

次節以降、(1) 組み込みボード IO 模擬ツール、(2) 機器間通信模擬ツールについて、詳細を述べる。

4.2 組み込みボード IO 模擬ツール

組み込みボード IO 模擬ツールは、CPU エミュレータ及び周辺デバイスエミュレータに接続し、各エミュレータの入力情報を設定し出力情報を表示する機能を有する GUI プログラムである。構成を図 5 に示す。

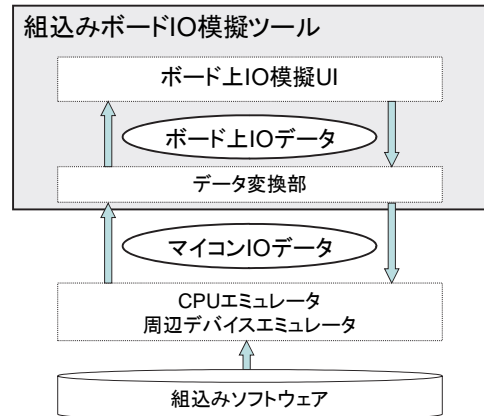


図 5 組み込みボード IO 模擬ツール及び周辺の構成

データ変換部は、各エミュレータが入出力するマイコンピン毎の入出力電位 (マイコン IO データ) と、組み込みボード上入出力インターフェース毎の人が認識容易な型データ (ボード上 IO データ) との、相互変換を実施する。

ボード上 IO 模擬 UI は、ボード上 IO データを実際の組み込みボードに近い見た目設定・表示可能な UI を提供する。開発したボード上 IO 模擬 UI と、対応するボード上 IO データについて、その一部を表 1 に示す。

表 1 ボード上 IO 模擬 UI (一部)

模擬対象 IO	I/O	ボード上 IO 模擬 UI	ボード上 IO データ
アナログセンサ	I		・センサ名、上限/下限値 ・データ型: 温度、圧力、照度など
H/L 入力バルス入力	I		・機能名 ・データ型: ON/OFF、H/L など
DipSW	I		・スイッチ名、数 ・データ型: ON/OFF など
LED	O		・LED 名、色 ・データ型: ON/OFF など
7SEG	O		・7seg 名、桁数 ・データ型: 数値、文字列 など
モータ	O		・モータ名 ・データ型: 回転数 (rpm)、正転/逆転/停止 など
外部ROM	O		・メモリ 番地、サイズ ・データ型: 数値 (HEX)

組込みボード IO 模擬ツールでは、これらのボード上 IO 模擬 UI を、機器毎にタブにまとめて表示させている (図 6) . 組込みボード IO 模擬ツールの実際の画面を、図 7 に示す.

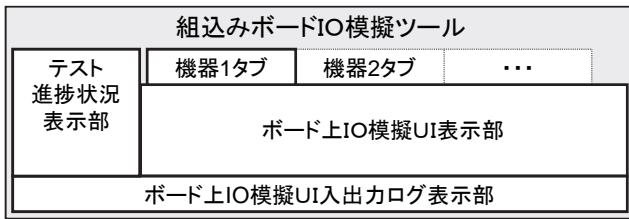


図 6 組込みボード IO 模擬ツールの画面構成

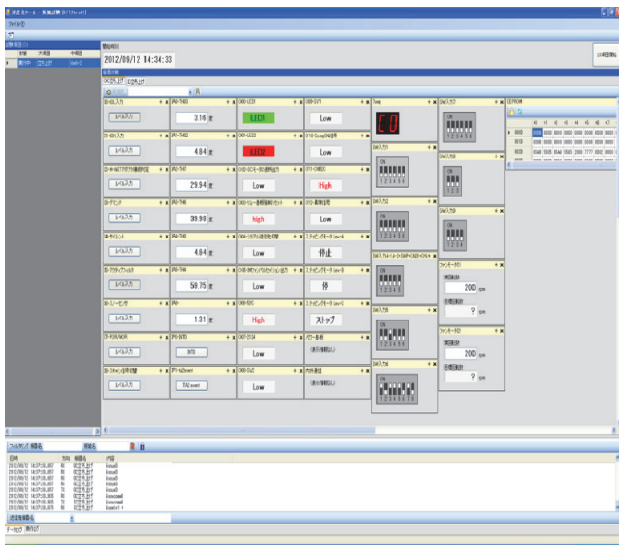


図 7 組込みボード IO 模擬ツールの画面

組込みボード IO 模擬ツールは以上の機能をもって、実機と同じ感覚で各エミュレータの入力設定・出力確認を実施できるようにしている。これにより、3 章で課題としたテスト環境の教育期間を短縮するとともに、テスト環境の操作・設定ミスを低減することが可能である。

4.3 機器間通信模擬ツール

機器間通信模擬ツールは、図 4 (3) インターフェースのドライバを含み、複数の CPU エミュレータ及び周辺デバイスエミュレータが入出力する機器間通信ネットワーク上コマンドの送受管理を行うプログラムである。構成を図 8 に示す。

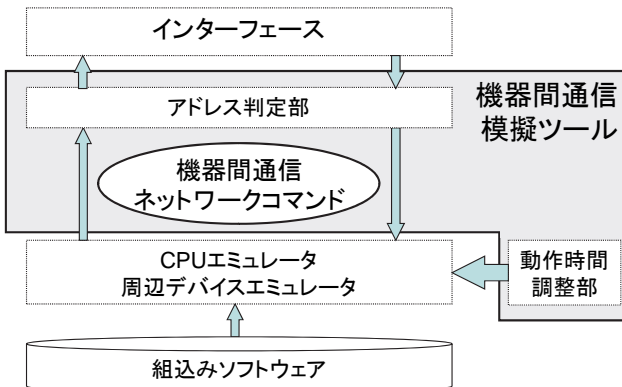


図 8 機器間通信模擬ツール及び周辺の構成

動作時間調整部は、各エミュレータの動作速度を、パソコン OS のタイマ精度の範囲で (クライアント向け Windows の場合、約 10ms) 実時間に同期させる (図 9) . この機能の実現には、オープンソース (二条項 BSD ライセンス) の CPU エミュレータである Softgun[3]を活用している。

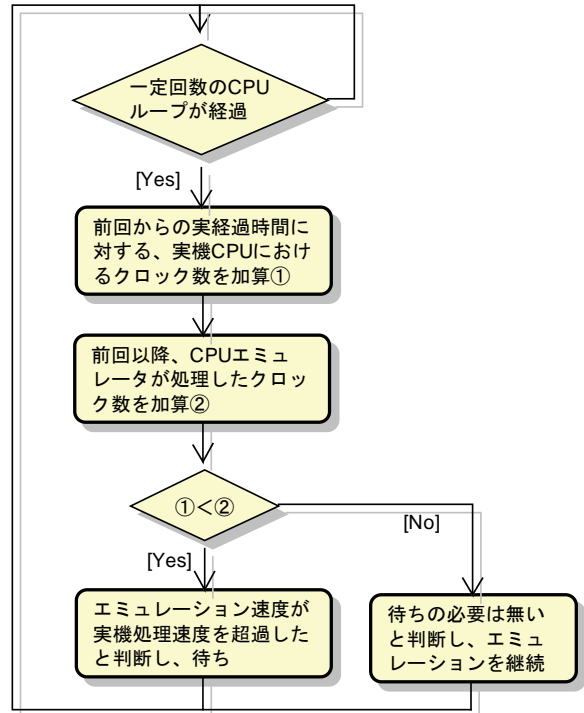


図 9 動作時間調整部のフローチャート

アドレス判定部は、各エミュレータ及びインターフェースが出力する機器間通信ネットワーク上コマンドの送信元アドレスを抽出し記録することで、各エミュレータのアドレスと、インターフェースを介して接続されている実機のアドレスとを認識する。そして機器間通信ネットワーク上コマンドの送信先アドレスを抽出し、記録していたエミュレータのアドレスと合致した場合、そのエミュレータにコマンドを転送する。

機器間通信模擬ツールは以上の機能をもって、実機とエミュレータとの間で機器間通信を実現する。これにより、図 4 に示した通り、被試験機は実機・試験機はエミュレータと言う組合せでのシステムテスト環境が構築可能になるので、3 章に示した製品の信頼性に関する課題を解決することができる。

4.4 評価

これまでに述べたシステムテスト環境が、3 章に記載の課題を解決できているかどうか、また本来の目的であるテスト作業の効率化を実現できているかどうかを評価するため、従来のシステムテスト環境及び開発しているシステムテスト環境を用いて実際のシステムテストを実施した。

実施したシステムテストの概要を、表 2 に示す。なお、システムテスト環境の構成は、図 3 及び図 4 に示した通りである。

表 2 実施したシステムテストの概要

テストの内容	設備機器システムの機器に対する、同時運転機能(他の機器の運転状態に合わせて、自身の運転状態が変化する機能)のテスト	
作業実施人数	3名	
作業者のスキル	従来システムテスト環境での作業、開発したシステムテスト環境での作業、双方の経験あり	
テスト実施回数	従来テスト環境: 10回 開発テスト環境: 10回 (毎回、試験機側の機器構成を変更)	
テスト作業詳細		
No	作業項目	作業概要
1	システム構成の構築	機器間をネットワーク接続する
2	初期状態の構築	機器の動作パラメータを設定し、システムを起動する
3	試験の実施	試験機の運転状態を切り替える
4	結果の確認	被試験機の運転状態を確認する

このテストの可否結果とともに、作業項目毎の作業時間を取得し、評価の材料とした。以下に評価結果をまとめる。

- 製品の信頼性確保についての評価
従来テスト環境と開発テスト環境とで、毎回同じ可否結果が得られた。
この事から、開発したテスト環境は、操作という非同期に発生するイベントについてはテストを実施可能であり、要求される信頼性の確保を実現できると考える。
- 導入の容易性についての評価
従来テスト環境と開発テスト環境とで、作業 No1～作業 No4 の内容を全く変えずとも、作業者はテストを実施することができた。
この事から、開発テスト環境は、従来テスト環境と同様にシステムの操作・状態確認ができ、現場にスムーズに導入できると考える。
- 作業効率化についての評価
従来のシステムテスト環境に対し開発しているシステムテスト環境がどれほどの作業時間短縮を実現したか、作業項目毎にまとめた。結果を表 3 に示す。

表 3 評価結果 (作業時間)

No	3名*10回のテストにおける作業時間の平均値(分)		作業時間短縮割合 (%)	t検定 (P(T<=t) 両側)
	従来テスト環境	開発テスト環境		
1	5.72	2.10	63	0.008(<=0.05)
2	4.17	1.90	54	0.033(<=0.05)
3	0.10	0.10	2	0.840(>0.05)
4	0.50	0.42	17	0.475(>0.05)
計	10.48	4.52	57	0.018(<=0.05)

作業全体としては 57%の効率化であり、t 検定によっても従来テスト環境における作業時間と開発テ

スト環境における作業時間との間に有意差があることが示されたことから、開発しているシステムテスト環境が、一定の作業効率化性能を有していることを確認できた。

特に、作業 No1, 作業 No2 について大きな効果が確認された。これは、テスト対象の機器に対応する組込みボードの調達・組込みボード間の配線・組込みボード上のスイッチ切替・と言った組込みボードに関わる作業が、全てパソコン上で完結するようになったためと推測する。

一方、作業 No3, 作業 No4 については効果は大きくなかった。これは 2 つの作業がともに短時間で完了するもので (No3 についてはスイッチ 1 つで切り替え可能, No4 については結果が一目で確認可能), 組込みボード上の作業とパソコン上の作業で負荷に差異が生じなかったためと推測する。

5. あとがき

本論文では、組込みシステム開発に対する CPU エミュレータの活用状況、及び設備機器システム開発への適用における課題 (製品の信頼性確保・開発現場へのスムーズな導入) について述べた。

そして、その課題の解決策として開発している CPU エミュレータを活用したシステムテスト環境が、これらの課題を解決しつつ、システムテスト作業の効率化を実現できることを示した。

今後は、以下に対応していく。

- 対応可能な機器間通信の拡張
組込みシステムの機器間通信には、パソコン OS のタイマ精度以上の時刻合わせ精度を必要とするものがある。このような機器間通信においても、実機とエミュレータ間での通信を実現できるようにする。
- さらなるシステムテスト作業効率化策の実装
組込みボード IO 模擬ツールを改善し、システムテスト作業をより効率化する。試験機構成の設定作業、ボード上 IO データの入力作業、等に改善の余地があると考えている。
- 実開発への適用
本システムテスト環境について、開発現場での評価を進め、適用可能なシステムテストの範囲・作業効率化性能を詳細化し、実開発への適用に繋げる。

引用文献

1. QEMU, a Fast and Portable Dynamic Translator. **BellardF.**: Proceedings of the USENIX Annual Technical Conference, FREENIX Track, pp. 41-46, 2005.
2. 仮想化技術による実機レステスト環境の構築. 原嶋秀次, 蔭山佳輝, 河込和宏. 2012年, 東芝レビュー Vol67 No8.
3. Softgun. <http://softgun.sourceforge.net/>.