

# IoT技術を用いた組み込み系製品の Raspberry Piを使用した自動テストの 問題点と解決方法

---

株式会社AGEST

ITソリューション営業本部 テストオートメーション営業部

○ 林 尚平

E-mail: [shohei.hayashi@agest.co.jp](mailto:shohei.hayashi@agest.co.jp)

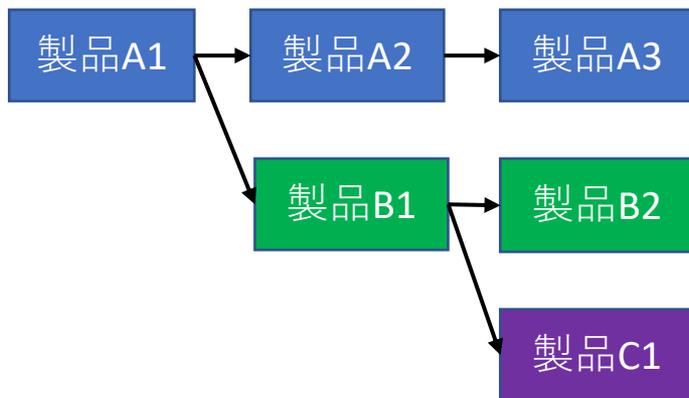
- **組込み系での自動テスト導入の背景・問題点**
- **自動テストの導入前の準備・調査**
- **自動テストを導入する際に行った対策**
- **自動テストの導入後の考察**
- **課題解決に向けた対策**

- **組込み系での自動テスト導入の背景・問題点**
- 自動テストの導入前の準備・調査
- 自動テストを導入する際に行った対策
- 自動テストの導入後の考察
- 課題解決に向けた対策

# ■ 組込み系製品の自動テストの必要性

業務系システムと同じく組込み系製品も品質確保とリリース期間短縮などを目的とした自動テストの導入検討が行われています。

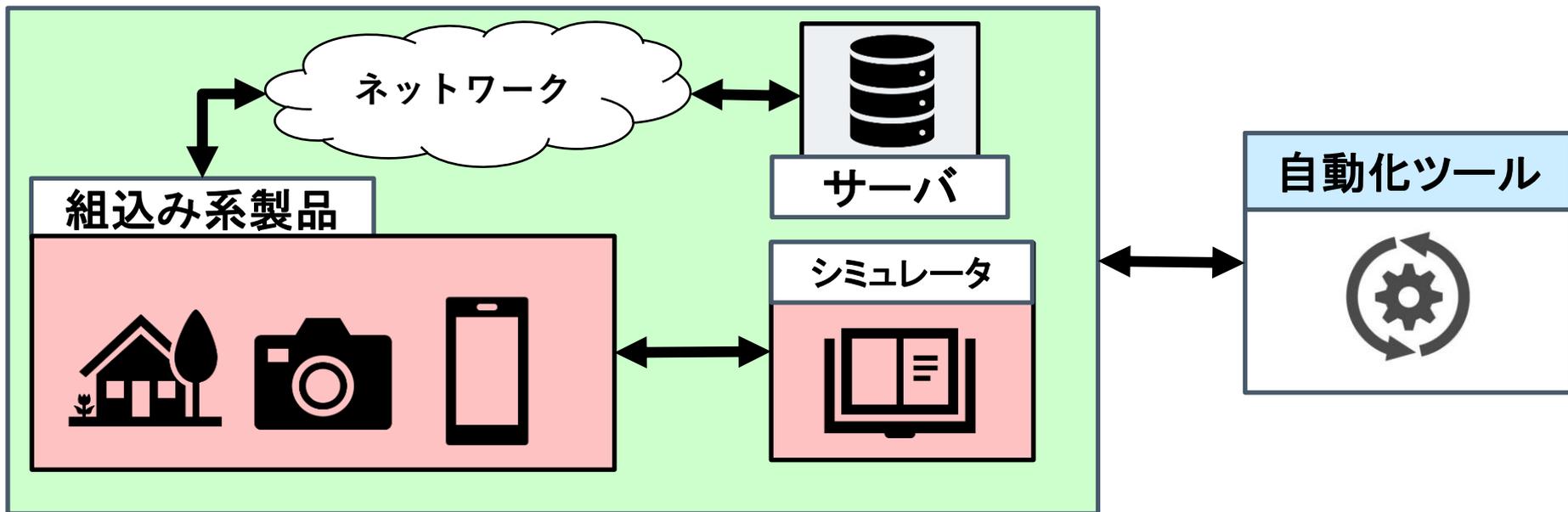
組込み系は派生機種が多い



派生機種が多くソフトリリースとなった際は評価工数が膨大になる

安心・安全にかかわる機能もあるため、評価は手が抜けない

# IoT技術を用いた組み込み系製品の自動テストの特徴



IoT機器の自動テストを行う場合は製品・サーバとの通信や連携するシミュレータ機器と自動化ツールを連携させる必要があります。

Raspberry Piを用いれば製品操作や電源制御、シミュレータなど操作などができるため、それを使った自動化ツールを作成することで自動テストが可能になります。

# ■ 業務系と組込み系の自動テストの違い

自動テストを行う際の大きな違いは自動化ツールです。

業務系システム	組込み系システム
PC内で処理が完結する	製品やシミュレータなどPCと連携する機器が多い
自動化ツールの選択肢が多い	製品に合わせて自動化ツールを作成する必要がある

業務系と同等レベルの自動テストが成立するツールの開発する必要があります。  
製品に合わせた自動化ツールを一から開発することが難易度を上げている要因。

- 組込み系での自動テスト導入の背景・問題点
- **自動テストの導入前の準備・調査**
- 自動テストを導入する際に行った対策
- 自動テストの導入後の考察
- 課題解決に向けた対策

# ■ 組込み系自動化ツールの調査結果

準備された組込み系製品の自動化ツールを使い、受け入れテストが自動化できるか確認したところ、以下の問題が判明しました。

No	問題点	詳細
①	前提条件の設定が自動化できない	試験の条件設定ができず、前シナリオの状態が引き継がれる
②	結果確認が自動化できない	試験手順を実行できるが、実行結果の判定ができない
③	自動化する機能が不足している	他ツール等との連携、テスト実施に必要な手順が自動化不可

この自動化ツールで製品操作、電源制御などの重要なテストの手順は自動化できるが、**テストの前提条件設定や期待結果の確認が自動化できない**ことが判明。

しかし他の自動化ツールでは代用できない重要な操作を自動化できるため、このツールを使い自動テストを成立させる必要がありました。

# ■ 自動テストの成功基準

自動テストの導入を行うために成功基準を定めました。  
業務系の自動テストと同等の効率化を目標とします。

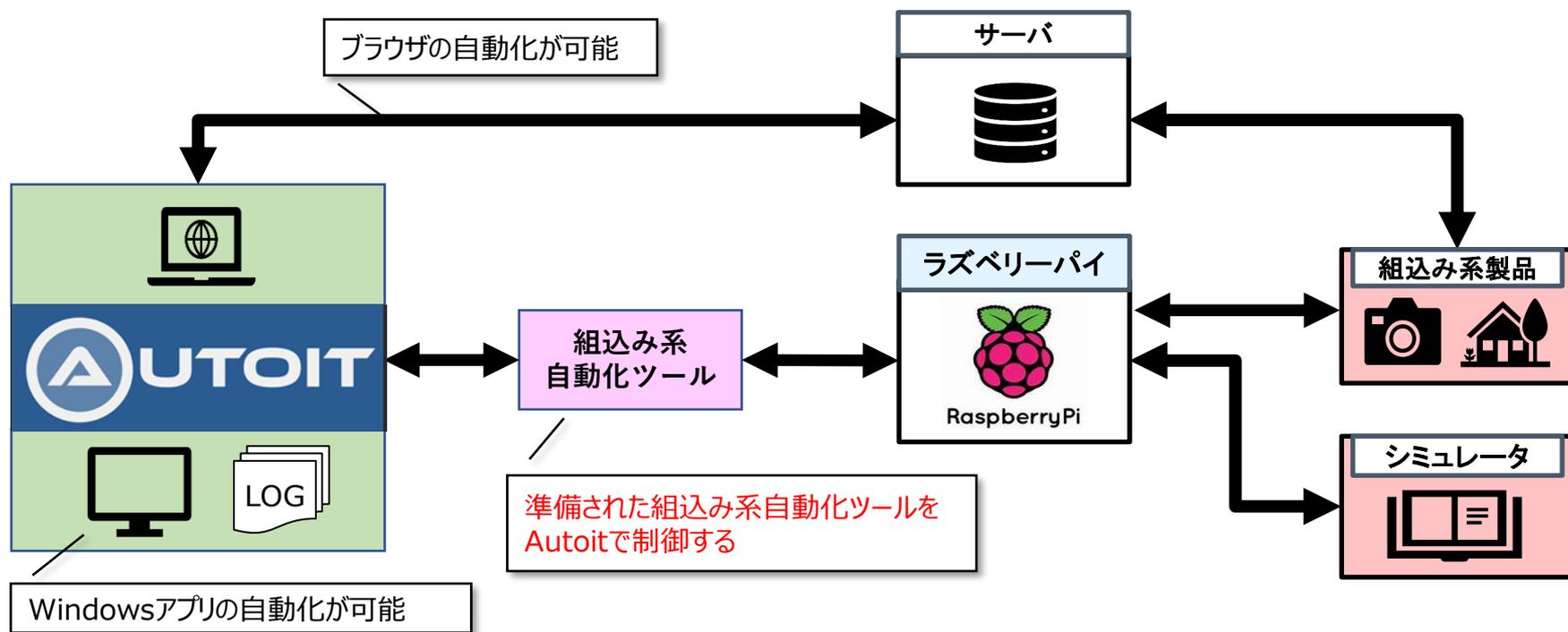
システムテストの受入れテストを  
自動化する場合の成功基準の例です

No	成功基準
①	手動に比べ30%以上の効率化/工数削減ができること
②	必要なテストが自動化できていること
③	実行途中で処理が止まらないこと
④	実行結果の誤判定がないこと
⑤	実行から結果確認まで自動化できること
⑥	共通関数などメンテナンス対策を行っていること

- 組込み系での自動テスト導入の背景・問題点
- 自動テストの導入前の準備・調査
- **自動テストを導入する際に行った対策**
- 自動テストの導入後の考察
- 課題解決に向けた対策

# ■問題に対する対策 2つのツールを連携させる

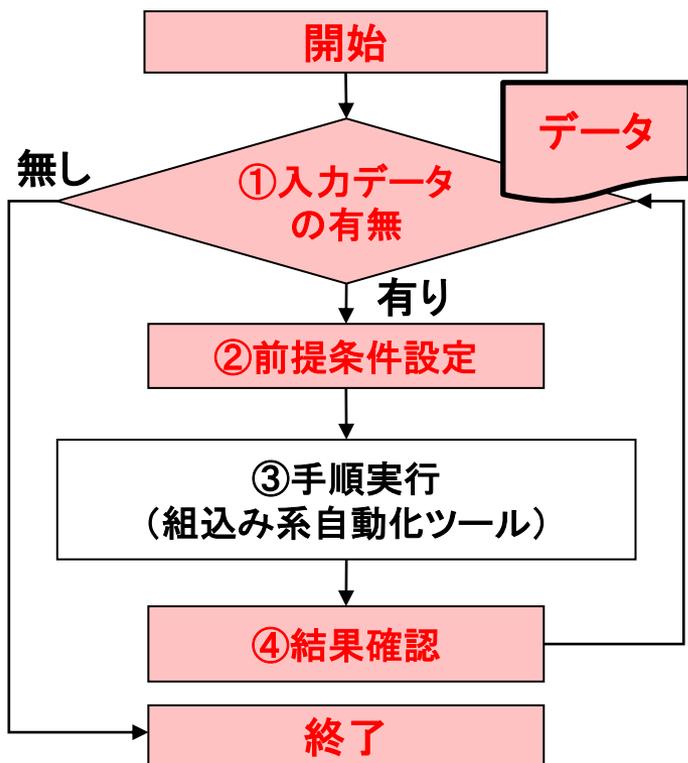
Autoitと組み系自動化ツールを連携させ、不足している処理を補い、必要な処理を全て自動化することで自動テストを成立させる。



**【組み系自動化ツール】**組み系製品、シミュレータの試験手順を自動化  
**【Autoit】**前提条件設定、結果確認、ブラウザ操作、その他の処理を自動化

# ■問題に対する対策(1) 全体処理をAutoitで行う

不足している他の処理を補うため、全体処理をRPAツール(Autoit)で自動化。  
組込み系自動化ツールは全体の一部の処理として呼出して実行する。

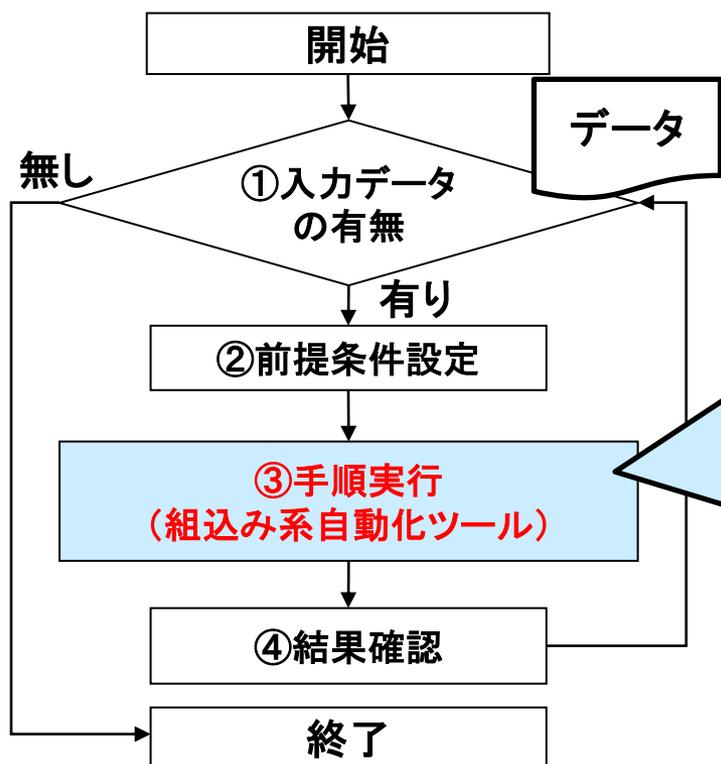


実行シナリオ	前提条件	ログ種別	期待結果
シナリオ_1	条件A	製品ログ	応答値_A
シナリオ_2	条件B	通信ログ	通信ログB
シナリオ_3	条件C	サーバログ	応答確認
シナリオ_4	条件D	製品ログ/サーバログ	応答値_B/サーバ疎通確認

2つの自動化ツールを連携すると  
パラメータを渡すことができませんが  
ファイルを仲介することで可能にしました

## ■問題に対する対策(2) 不足機能の追加

組込み系自動化ツールを呼出し中の処理はAutoitでは制御できないため、組込み系自動化ツール内で作成した実行ファイルを呼び出す。



### 【組込み系自動化ツールのシナリオ内容】

電源: ON

音楽画面へ遷移

**実行: 作成した実行ファイル**

音楽再生

音楽停止

ホーム画面へ遷移

- 組込み系での自動テスト導入の背景・問題点
- 自動テストの導入前の準備・調査
- 自動テストを導入する際に行った対策
- **自動テストの導入後の考察**
- 課題解決に向けた対策

# ■ 自動テスト導入の効果

自動化に成功し派生機種を含めて数十機種の製品の自動テストの導入ができ、ソフトリリース対応の効率化が可能になりました。

No	自動テスト導入効果
①	1機種当たり受け入れテストの実施で40%~80%の効率化が可能
②	フリーのRPAツールを使用したことで、自動化環境が増えてもツールのコスト不要
③	機種数が増えても同じ自動化システムを使える
④	夜間に実行することで人間の作業が大きく軽減された

リリース作業の大幅な効率化を実現できたため、自動テストの導入は成功。

# ■ 自動テスト導入後の成功基準の考察

自動テストの導入ができたので導入前に定めた成功基準の考察を行いました。

No	導入基準	結果	結果詳細
①	手動に比べ30%以上の効率化/工数削減ができること	◎	40%～80%の効率化が可能
②	必要なテストが自動化できていること	○	最大70%の機能は自動化可能
③	実行途中で処理が止まらないこと	△	設定ミスで実行エラーが頻発
④	実行結果の誤判定がないこと	△	設定ミスで実行エラーが頻発
⑤	実行から結果確認まで自動化できること	◎	一連の動作は全て自動化できている
⑥	共通関数などメンテナンス対策を行っていること	◎	自動化に必要な対策は完了

複雑な構成のため **設定ミスにより実行エラーが多発**  
構成の改善や実行前のチェックを行ったが改善に至らず。

# ■運用上の問題点

運用上に問題が発生することが判明しました。  
設定ミスによる要因は以下になります。

No	実行エラーの要因	詳細
①	機種数の多さ	数十機種を超えるためそれぞれの機種の設定が必要
②	組込み系製品の外機器、サーバとの連携	他機器やサーバとの連携の設定ミス
③	自動化ツール2つ使うことによる連携ミス	複雑な仕組みになってしまうため、パラメータ連携ミス
④	自動化システムの設定ファイルミス	設定ファイルによる機種、ツールの格納先などの設定ミス



**2つの自動化ツールを使うことで複雑な仕組み**になっていることが、設定ミスを生む大きな要因となっています。  
運用上問題になるため対策が必要だと考えます。

- 組込み系での自動テスト導入の背景・問題点
- 自動テストの導入前の準備・調査
- 自動テストを導入する際に行った対策
- 自動テストの導入後の考察
- **課題解決に向けた対策**

# 組込み系自動化で問題解決に必要な条件

## ■運用を考えた自動テストの条件

- ・理解しやすいシンプルな構造であること
- ・1つツールで全て自動化できること

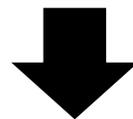
条件が複雑になる要因

- ・機種が多い
- ・連携する機器が多い

## ■自動化ツールの必要な条件

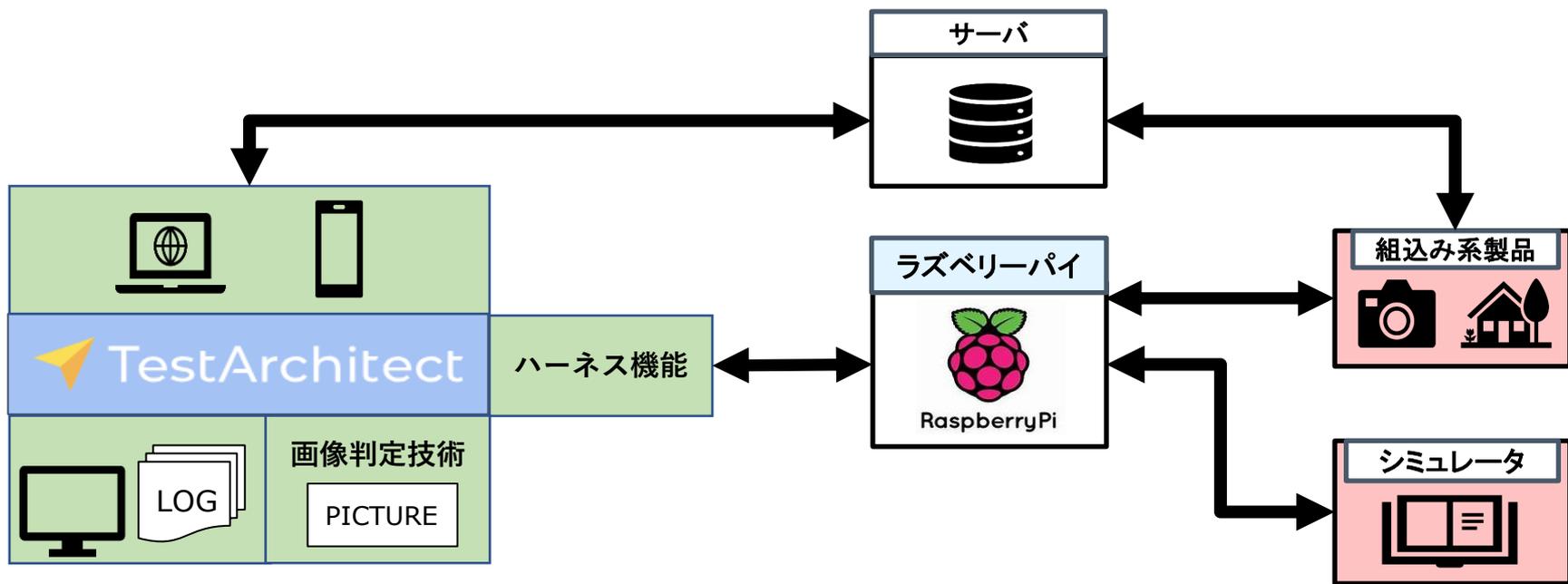
- ・組込み製品や試験に必要なシミュレータを制御できること
- ・Webブラウザ、モバイルなどサーバとの連携ができる機能があること
- ・豊富な機能で前提条件設定、手順実施、期待結果の自動化ができること

テストに必要な機能は  
全て自動化できること



**AGESTの製品であるTestArchitectを用いることで、上記の条件が解消。  
その自動化システムの概要について次ページにて紹介**

# TestArchitectを使用したテスト自動化の概要



TestArchitectの機能	特徴
基本機能	充実した機能を持った自動化ツールで細かい処理も自動化可能
わかりやすい自動化システムの構造	ツールが1つで運用できるため、設定ミスの減るスマートな構造
ハーネス機能	ラズパイを制御する機能を使い製品とシミュレータをスクリプト上で自動化
モバイル、ブラウザ機能	サーバと連携が可能
OCR/Key Point技術による画像判定	期待結果の画像の正確な判定が可能

TestArchitectを用いることでIot機器の全対象機材の自動テストが実現可能

# まとめ

組込み系製品の自動化テストの注意点は以下になります。

No	組込み系の自動テストの注意点
①	運用を考えた自動テストを設計を行うこと
②	複数のツールを連携させず、1つのツールで自動テストを実現すること
③	できる範囲だけの自動化にせず、テストに必要なもの全てを自動化すること
④	コスト削減だけに着目せず必要なコストは掛けること
⑤	自動テストの知識や経験を持った技術者が必要

運用を考えた自動化設計とそれを実現する自動化ツールの選定がIoT機器の組込み系製品の自動テストには重要です。それには自動テストの知識が必須です。

---

ご清聴ありがとうございました