

IoT プラットフォームを用いた機器制御アプリケーションの実装

森 郁海[†] 鷲尾 元太郎[†] 田村 孝之[†]三菱電機株式会社 情報技術総合研究所[†]

1. はじめに

近年、様々なモノに通信機能を持たせ、インターネットを経由し相互に通信することにより、遠隔計測や自動制御等を実現する IoT(Internet of Things)/M2M(Machine to Machine)が注目されている。IoT の用途は、(1)監視、(2)制御、(3)最適化、(4)自律性の 4 段階を経て発展する[1]。IoT システムの大規模・複雑化に伴い、各機器の振る舞いの違いが課題となる。アプリケーションが個別に機器の差異を吸収するのは好ましくなく、機器に対する抽象的なインタフェースをアプリケーションに提供する IoT プラットフォームが求められる。

本稿では、IoT プラットフォームを用いた機器制御アプリケーションの実装例を挙げ、機器の振る舞いの差異への対応の重要性を示す。

2. 課題

機器の振る舞いの差異の例を図 1、図 2 に示す。機器に直接命令するタイプのもの(図 1)では、アプリケーションからの命令を受けてから処理を実行するので、命令の結果と実際の機器の状態が乖離することがないが、機器の命令処理性能がシステムの応答性能に直接影響する。

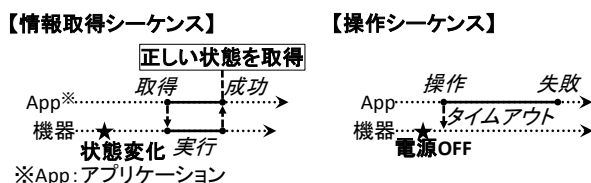


図 1 機器の振る舞いの差異(機器直接命令)

一方、中継器を介すタイプのもの(図 2)は、応答性能を高めるために中継器が以下のような処理を実行する場合がある。

- (1) 機器の状態をポーリングし、アプリケーションからの要求を受けた時点で即時応答する
- (2) 操作命令を受けた時点では機器を操作せず、命令を受信した旨を即時応答する

(1)では、ポーリング間隔によっては、実際の

機器の状態と異なる状態が取得され、システムの挙動に異常をきたす場合がある(図 2 情報取得シーケンス)。

(2)では、操作命令の応答だけでは機器への操作の成否を判断することができない問題がある(図 2 操作シーケンス)。

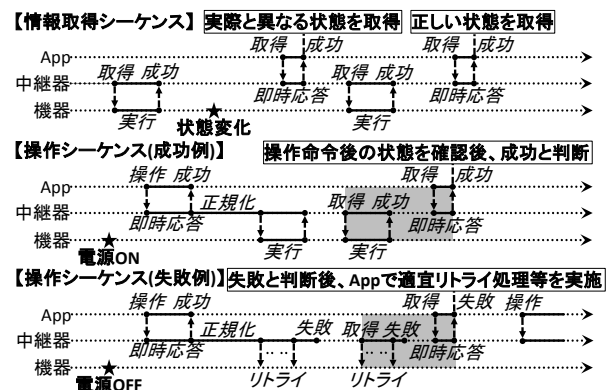


図 2 機器の振る舞いの差異(中継器経由)

3. 機器の振る舞いの差異の吸収方式

本機器制御アプリケーションでは、アプリケーションと機器の間に IoT プラットフォームを配置する。制御アプリケーションは、IoT プラットフォーム経由で情報を交換することで、アプリケーション-機器間のネットワーク構成を意識することなく、機器と通信できる。IoT プラットフォーム内部の通信は、ロングポーリング[2]を利用している。なお、通信プロトコル部分は、IoT プラットフォームによって隠蔽されており、アプリケーション側では通信プロトコルを意識することなく、通信機能を利用可能である。

2. 課題で述べた(1)に起因する問題を解決するには、中継器と機器間のポーリング間隔を短くする必要があるが、今回使用した中継器にはポーリング間隔を設定する API が存在しなかった。前記の対策を講じて、機器の状態が変化してからポーリングする間にアプリケーションから情報取得要求を受けると状態が乖離してしまう。根本的には、中継器が要求を受けてから機器の情報を取得するものを使用するか、運用でカバーする必要がある。

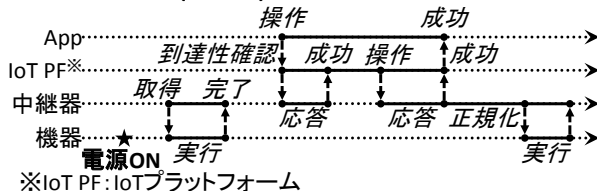
Implementation of Device Control Applications on an IoT Platform

[†] Ikumi MORI, Gentaro WASHIO, Takayuki TAMURA

Information Technology R&D Center, Mitsubishi Electric Corporation

(2)に起因する問題は、機器への到達性の確認を実施してから操作命令を実行することで解決できる(図 3)。到達性の確認は、(1)に起因する問題を含むが、命令する直前の機器の状態を確認することで、操作命令が成功する確率が上がる。

【操作シーケンス(成功例)】



【操作シーケンス(失敗例)】

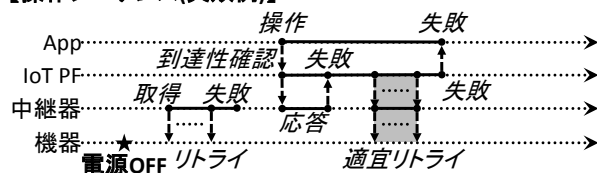


図 3 課題解決法の一例

4. 評価と考察

IoT プラットフォームの応答性能を評価するために、機器の操作/情報取得の所要時間を計測した(表 1)。機器に直接命令するタイプのものとして、ネットワーク表示灯(パトライト社製 NHL-5FV1)、中継器を介すタイプのものとして、照明(Philips Hue)を使用した。表中の値は、10 回の計測値の平均である。測定は、制御アプリケーションに測定用コードを埋め込んで実施した。参考値として、アプリケーションとネットワーク表示灯間の ping の所要時間を併記する。

表 1 評価結果

		所要時間
ネットワーク	機器操作	129 ms
	機器情報取得	1011 ms
照明	機器操作	1122 ms
	機器情報取得*	821 ms
アプリ⇄IoTプラットフォーム間通信(往復)		15 ms
Ping(アプリ⇄ネットワーク表示灯)		2 ms

※ 到達性確認にも使用

表 1 から IoT プラットフォームの通信が占める割合は、機器の操作/情報取得の所要時間に対して少ない(1.3~11.4%)ため、機器の命令実行処理がボトルネックとなっている事がわかる。

ネットワーク表示灯の場合、操作より情報取得の所要時間が長い。これは、機器内部の操作処理負荷より状態取得処理負荷の方が高いためである。

一方、照明の場合、情報取得より操作の所要時間が長い。情報取得では、要求を受けた時点でポーリングデータを即時応答するが、操作では、要求を受けてから制御を実行するためである。また、本稿での操作処理は、3. 提案方式で述べたように到達性確認後に実行されるので、制御アプリケーションとしての操作成功時の所要時間は 1943ms(821ms+1122ms)となる。情報取得については、機器の電源を ON→OFF にした場合、正しい状態が取得できるまで最大 30 秒程度要した。

以上のことから、機器の応答性能や中継器の API の仕様(実際の機器状態との乖離等)がシステムの挙動に強く影響することがわかる。特に、機器の情報取得のポーリング間隔のように IoT プラットフォームでは吸収できないものがあるため、システムの要求条件を満たせない場合には、運用でカバーするか、要求条件を満たせる機器に交換する必要がある。

5. まとめ

本稿では、IoT プラットフォームを用いた機器制御アプリケーションの実装例を挙げ、機器の振る舞いの差異への対応の重要性を示した。

機器に直接命令するタイプでは、機器の応答性能がシステムの応答時間に直接影響するため、要求定義を満足する応答性能を持つ機器を設計段階から選定する必要がある。

中継器を介すタイプでは、中継器-機器間のポーリング処理により、中継器の API から取得した情報と実際の機器の状態の乖離が発生するため、システムの要件に合わせた対応(機器の再選定や運用でのカバー等)が必要となる。

今後は、IoT プラットフォームの適用機器の拡充や、機器共通機能のライブラリ化等に取り組んでいく。

参考文献

- [1] Michael E. Porter, James E. Heppelmann: How Smart, Connected Products Are Transforming Competition, HBR, November 2014
- [2] D.Guinard et al.: "Cloud Computing, REST and Mashups to Simplify RFID Application Development and Deployment", WoT 2011, pp.1-8 (2011-6)