

モバイルアプリケーション活用による 生産現場の見える化ソリューション

原 泰裕*
Yasuhiro Hara
兼子貴弘*
Takahiro Kaneko

Production Site Visualization Using Mobile Applications

要 旨

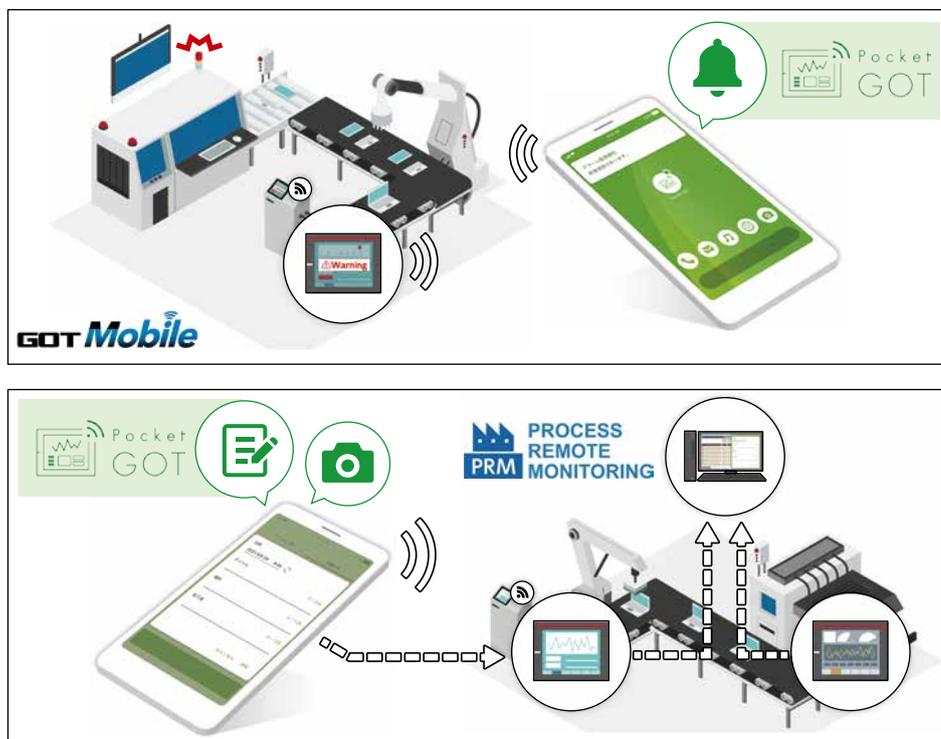
近年、工場の生産現場では省人化で一人の作業者が担当する装置が増加し一つの装置に作業者が常駐することが減少している。さらに新型コロナウイルスのまん延を受け、リモートワーク化が進み、低コストで容易に導入できるリモートソリューションの要求が増加している。同時に設備のIoT化に伴い、リモートソリューションを導入する上で、設備から情報を収集する“生産現場の見える化”が求められている。

これらの現状を踏まえ、リモートワークも視野に入れた三菱電機製タッチパネル付き表示器“GOT2000シリーズ”のリモートソリューションである“GOT Mobile”と連携した

モバイルアプリケーション“Pocket GOT”を開発した。これによって簡単に設備のIoT化を実現できるFAアプリケーションパッケージ“iQ Monozukuri工程リモート監視”^①を活用した生産現場の見える化ソリューションを実現した。

Pocket GOTの特長は次のとおりである。

- (1) 生産現場で発生したアラームを受信し、バイブレーション、音、バナーなどで通知する。
- (2) GOT Mobileと連携し、アラームが発生した現場情報の確認ができる。
- (3) モバイル端末の写真や画像、テキストで登録した現場情報の共有ができる。



モバイルアプリケーション“Pocket GOT”

離れた生産現場で発生したアラームを手元のモバイル端末へ通知し、従来のリモートソリューションであるGOT Mobileと組み合わせることで、異常発生時の現場での対応が更に迅速になる見える化を実現するモバイルアプリケーションである。

1. ま え が き

GOT2000シリーズ(以下“GOT”という。)は、“Easy & Flexible(使いやすく、自由度が高い)”をコンセプトに開発された産業用のタッチパネル付き表示器であり、様々なFA機器と接続でき、装置全体の見える化を行えることから国内外で高い評価を得ている。そのGOTを通じて、遠隔地のパソコンやタブレットなどのモバイル端末から現場の接続機器をモニタ・操作を可能にする当社製リモートソリューションであるGOT Mobile(図1)が市場で好評であったことを受け、機能開発への需要が更に高まった。

一方で、GOT Mobileは遠隔地のパソコンやモバイル端末でWebブラウザを起動していないと現場で異常が発生したことを迅速には把握できないため、常にWebブラウザを起動しておく必要があった。

そこで離れた生産現場の状況をより迅速に把握するため、生産現場で発生したアラームの通知を手元のモバイル端末で受信し、モバイル端末やパソコンから設備の遠隔監視・操作を可能にするGOT Mobileと連携したモバイルアプリケーションPocket GOTを2021年5月に市場投入した。

本稿では、Pocket GOTの特長、機能実現のための技術と工夫した点について述べる。

2. Pocket GOTの特長

2.1 リモートで異常を把握

Pocket GOTは、Android^(注1)版のモバイルアプリケーションGoogle playで無償公開している。Pocket GOTは、離れた生産現場のGOTから送信されたユーザーアラーム発生情報を手元にあるモバイル端末で音、振動、バナーによって受信できる。このため、作業者は従来のようにモバイル端末でWebブラウザを起動しておかなくても生産現場の異常を迅速に把握できる(図2)。

(注1) Androidは、Google LLCの登録商標である。

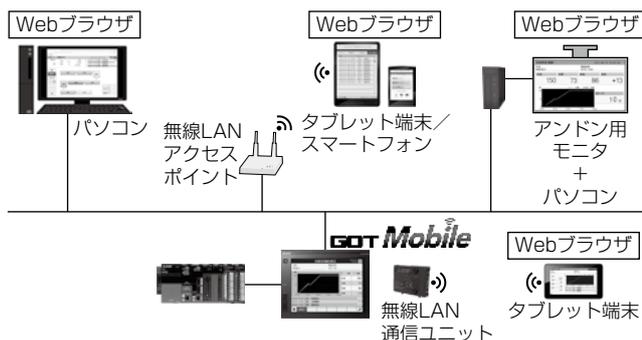


図1. GOT Mobileのシステム構成

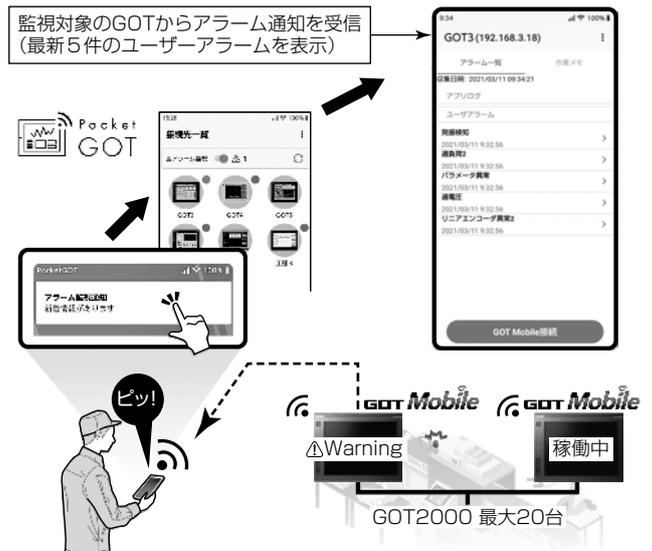


図2. リモートでの異常把握

2.2 設備状態とアラーム履歴を確認

当社のリモートソリューションGOT Mobileと連携してPocket GOTからGOT Mobileで使うWebブラウザを起動してGOT Mobile専用画面を表示できる。このため、アラーム通知した現場の情報やアラーム履歴を手元のモバイル端末で確認できる(図3)。

2.3 作業メモの一括表示

Pocket GOTをインストールしているモバイル端末で作成した画像付きの作業メモ(日報や報告書など)をGOTに送信できる。また専門的な知識がなくても簡単にシステム構築が可能で、設備のIoT化を実現できる当社製FAアプリケーションパッケージiQ Monozukuri工程リモート監視を活用して生産現場の各ラインに設置されているGOTから作業メモの収集が可能である。これによって事務所のパソコンで帳票形式の作業メモを一括で表示でき、写真や画像、テキストで登録した現場情報を共有できる(図4)。



図3. 設備状態とアラーム履歴の把握



図4. 作業メモの一括表示

3. Pocket GOTの開発

3.1 アプリケーション形態

モバイルアプリケーションのアプリケーション形態には、アプリケーションのインストールが必要でアプリケーション単体で動作する“ネイティブアプリケーション”，アプリケーションのインストールが不要でWebブラウザ上で動作する“Webアプリケーション”，インストールが必要であるがメイン機能はWebブラウザ上でWebアプリケーションとして実行される“ハイブリッドアプリケーション”の三つが存在する。

Pocket GOTは、生産現場で発生したアラームを手元のモバイル端末で受信するために、常にモバイル端末のバックグラウンドでも動作する処理を実装する必要がある。しかしながら、Webブラウザ上で動作するWebアプリケーションやハイブリッドアプリケーションはバックグラウンドで実装できない機能があるため、バックグラウンドで実装する機能に制限のないネイティブアプリケーションを採用した(表1)。

3.2 開発環境

Pocket GOTをインストールするモバイル端末には、マルチプラットフォーム環境(iOS^(注3)、Android)への対応が求められており、Visual C#^(注4)やVisual studio^(注4)の知

表1. モバイルアプリケーション形態比較表

選定ポイント	ネイティブアプリケーション	Webアプリケーション	ハイブリッドアプリケーション	重要度
端末機能制約 ^(注2)	○：制約なし	×：制約あり	△：一部制約あり	↑ 高 ↓ 低
ネットワーク接続	○：不要	×：必要	△：一部必要	
システム構成	○：容易	×：複雑	×：複雑	
動作安定性	○：高い	×：環境依存	△：一部環境依存	
端末の汎用性	×：低い	○：高い	×：低い	
プログラムの汎用性	×：低い	○：高い	○：高い	

(注2) カメラ・通信機能など

見を活用できることや当社の他製品で実績があることからPocket GOTの開発環境としてMicrosoft社のクロスプラットフォーム開発環境であるXamarin⁽⁴⁾を採用した。

(注3) iOSは、Cisco Technology, Inc.の登録商標である。

(注4) Visual C#, Visual studio, Xamarinは、Microsoft Corp.の登録商標である。

4. 機能実現のための技術、工夫した点

4.1 システム構成

Pocket GOTのシステム構成を述べる。Pocket GOTは、離れた生産現場の異常を通知する“アラーム通知機能(図5①)”と生産現場の点検結果を作成する“作業メモ機能(図5②)”で構成されている。アラーム通知機能は、同一ポートを使ってサーバ/クライアント形式で通信するHTTP(HyperText Transfer Protocol)通信を活用しており、GOT Mobileで搭載実績のある通信ライブラリによってアラーム要求をHTTP通信で送信し(図5③)、要求を受けたGOT MobileはHTTP通信によってアラーム情報を通知する役割を担う(図5④)。作業メモ機能は、生産現場で記録した文字列や画像から構成されるデータをGOTへFTP(File Transfer Protocol)通信によって送信する役割を担う(図5⑤)。

またパソコンに複数設備の情報の収集と見える化をするFAアプリケーションパッケージiQ Monozukuri工程リモート監視(図5⑥)をインストールしておくことで、GOTから作業メモデータを取得し(図5⑦)、HTML(HyperText Markup Language)形式で一括表示する(図5⑧)ことを可能にした。

4.2 アラーム通知方法

アラーム通知方法には、モバイルアプリケーション内部で通知を発行することで通知をモバイル端末に表示するローカル通知と、インターネットへ接続することでクラウドサーバから発行される通知を受け取りモバイル端末で表

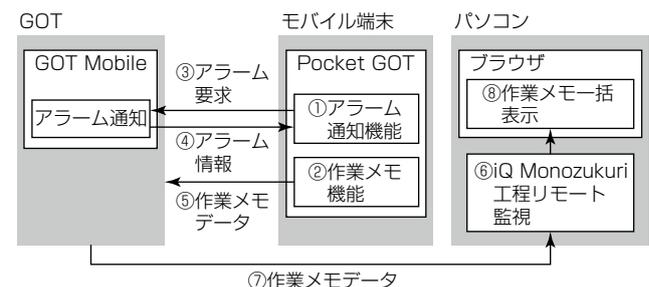


図5. Pocket GOTのシステム構成

示すオンライン通知がある。前者はオフライン端末への通知機能の実装量が少ない点がメリットだが、スリープ時の動作担保の難易度が高い点がデメリットである。後者は、バックグラウンドでの通知で実装が容易な点がメリットだが、インターネットへ接続されていないオフライン端末への通知の実装が不可であり、通知用のサーバが必須で構成が複雑になる点がデメリットである(図6)。

Pocket GOTでは、オフライン端末への通知が可能であり、容易に導入できるローカル通知で実現する手法を取り入れ、Pocket GOTからGOTへアラーム発生状況の要求を送信し、GOTからアラーム情報を受信し、新規アラームがあればPocket GOT内部で通知を生成するシーケンスを採用した(図7)。

4.3 モバイル端末スリープ時の通信処理の分割

モバイル端末がアクティブ状態の場合、フォアグラウンドとバックグラウンドの動作がある。フォアグラウンドは、モバイルアプリケーションの動作が優先される状態であり、基本的に動作に制限はない。バックグラウンドは、モバイルアプリケーションは起動しているが、モバイルアプリケーション画面が表示されていない状態であり、画面に

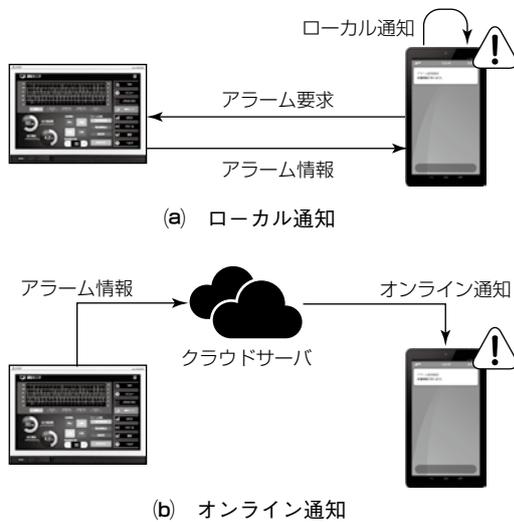


図6. ローカル通知とオンライン通知

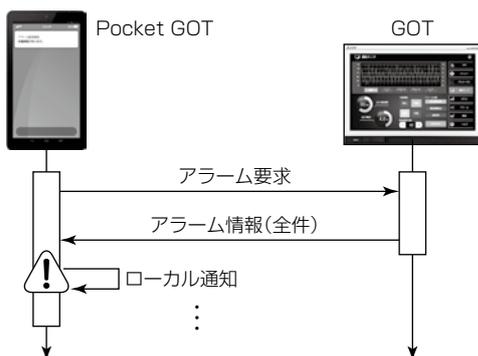


図7. ローカル通知のシーケンス

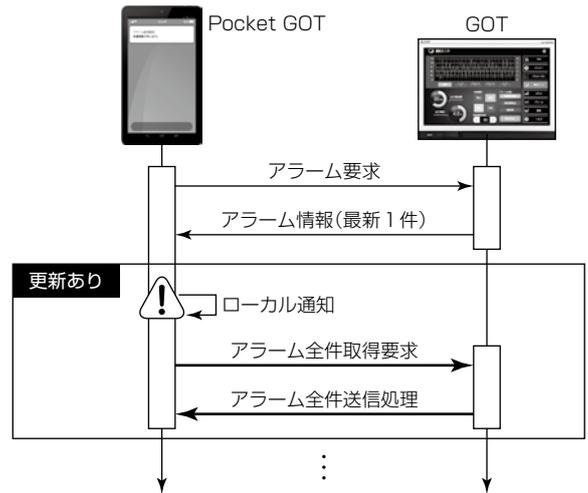


図8. 通信処理の分割

表示されている他のアプリケーションよりも動作の優先度が低いためモバイルアプリケーションの動作に制限がある。またモバイル端末がスリープの場合は、モバイルアプリケーションの動作に強い制限がある。

このような動作制限によって発生し得る様々なリスクを考慮し、Pocket GOTをバックグラウンドやスリープ時でも、処理としてはフォアグラウンドとして動作させる仕組みにした。しかし、モバイル端末がスリープ時にPocket GOTからGOTへ定期的にはアラームを要求する処理に遅延が発生する課題が顕在化した。

そこでPocket GOTは、GOTへ送信したアラーム要求に対して、GOTから最新1件のアラーム情報だけを受信し、Pocket GOT内部で前回受信時のアラーム情報と比較して通知処理が必要か判定する処理に変更することで、スリープ中の処理を削減し、周期性を確保した。この通信処理の分割によってスリープ時も含めバックグラウンドでのPocket GOTの処理を定期的に行うことができる仕組みを実現した(図8)。

5. む す び

GOT2000シリーズとGOT Mobile, iQ Monozukuri工程リモート監視との連携によって、異常発生時の現場での対応が更に迅速になる見える化ソリューションを生産現場で簡単・低コストで導入できるモバイルアプリケーションPocket GOTについて述べた。

今後は、ユーザーの使いやすさを更に追求し、機能拡充を行っていく。

参考文献

- (1) 大小嶋紗碧, ほか: FAアプリケーションパッケージ“iQ Monozukuri工程リモート監視”, 三菱電機技報, 94, No.4, 236~239 (2020)