

制御機器からの IoT データ収集技術

小林 義明

近年、センサなどの制御機器の生み出すセンシングデータをAIなどの最新IT技術を駆使して活用することの重要性が高まっている。その促進には制御機器の不統一なセンシングデータを体系化されたIoTデータに変換して収集・蓄積することが重要である。

FA現場で制御を担ってきたPLCをデータハブとすることで、多様な制御機器からの生データを体系的にIoTデータに変換することを試みた。

実現手段として制御機器の仕様情報をライブラリ化した「機器情報ライブラリ」と、それを参照してデータ収集・変換プログラムを自動生成するソフトウェア「フォーマットングツール」を開発した。自動生成されたプログラムをPLCに転送することでデータ収集から変換、IT層への送信を統合的に実行できる。

IoT Data Collection from Control Devices

Yoshiaki Kobayashi

In recent years, it has become important to make a full use of sensing data from control devices like sensors via new IT technology such as AI. To accelerate this, it is important to collect and transform ill-formed sensing data from control devices to well-formed data.

In this paper, we have tried to transform raw data from various control devices to IoT data systematically by using PLC in FA field as a data hub.

As an implementation, we have developed "Device Information Library" which contains specifications of devices and "Formatting Tool" software which generates PLC programs to collect and convert data based on Device Information Library. The generated programs transferred into a PLC execute integrally data collection, transformation and upload to IT layers.

1. まえがき

ファクトリーオートメーション（以下、FA）では用途やその時代の技術水準に応じて様々な制御機器インタフェースが乱立してきた。インタフェースの例としては古くからあるアナログ入出力や、近年ではIO-Linkなどの通信インタフェースがある。FA現場では対象や目的に応じてこれらの中から最適な機能・性能を持つインタフェースが選択されてきた。また、制御機器は長期間利用し続けることが珍しくなく、新規設備であっても信頼性重視の観点から旧式で実績のある制御機器をあえて選択する場合もある。この結果、FA現場では接続インタフェースが劇的に淘汰・集約されることはなく混在して稼働していることが一般的である。

このような多種・多様な制御機器を機能させるためには、制御機器（以下、単に機器）をプログラマブルロジックコントローラ（以下、PLC）や産業用PC（以下、IPC）のような汎用機器に接続したうえで、制御用のプログラムを記述するのが一般的である。プログラミングに際してはイン

タフェースごとに異なる知識が必要となる。また、同じインタフェースであっても機器ごとに単位や有効桁数などの特性が異なっている。さらにプログラミングは自由度が高いため、プログラミング結果は属人的なものとなる。このような多様性や属人化によりFA現場が生み出すデータ形式は不統一になりがちであり、制御データを分析などに二次利用する妨げとなっている。

従来の装置開発ではPLCやIPCなどのコントローラを中心とした機器間の連携が同一装置内に限られることが多いため、データ形式の不統一が問題視されることは少なかった。対象となる機器は制御に直接関わるものに限られ技術者の負担はそれほど大きなものではなかった。性能・品質の属人性は摺り合わせ型開発としてむしろ強みとも考えられてきた¹⁾。

しかし、第四次産業革命²⁾を目指す取り組みが加速する中、制御には直接関係しないものも含め、かつてないほどの多くのセンシングデータを収集して、装置内外を問わず横断的に分析活用する必要性が高まってきた。そのような活用の例としては総合設備効率や省エネルギー、トレーサ

ビリティなどがある。オムロンもIoTサービス基盤「i-BELT」を立ち上げ、この取り組みを加速させている³⁾。

このような用途ではデータを現場で収集された「あるがまま」の多様な形式で集めるのではなく、単位や精度などのメタ情報を付加したうえで、IT層での分析やAI処理などに適した形式に統一して収集することが求められる⁴⁾。

また、各データの有用性は分析をして始めてわかることが多く、収集対象を絞ることは困難である。このため、多数のセンサからデータ収集するためのプログラミングコストは増大し、情報活用の取り組みを開始する際の大きな障害となる。

本稿ではライブラリ化された機器の仕様情報に基づいてデータ収集・変換プログラムを自動生成することでPLCを機器から収集したデータを集約して送信するデータハブとし、アナログセンサを含めた多様な制御機器からの生データを体系的にIoTデータに変換する試みについて述べる。

2. 課題

2.1 OT-IT間のデータモデルの違い 制御機器から収集したデータをAIなどのIT技術を駆使して活用するためには、リレーショナルデータベースやデータレイク、メッセージブローカなどのIT技術に基づくストレージやインタフェースに送信することが一般的である。送受信のためにはPLCやIPCが送信するデータモデルと、受信するIT側のデータモデルを同一にすることが必要である。PLCやIPCのプログラムはOT (Operational Technology: 運用・制御技術) と呼ばれるスキルを持ったエンジニアによって作成されるのに対し、受信側はITスキルを持ったエンジニアによって作成される。

このようにOTとITの間のIoTのためのデータ交換では、異なる分野のエンジニア間でのコミュニケーションが発生する。ここで用語の違いや知識の違いがあるため、データ交換の実現までに時間がかかり、データ交換において齟齬が発生し誤ったデータが利用されるなど課題が生じやすい。

また、FA現場の装置はCADやPLCツールを使ってFA技術者によって設計され、機器の接続などの構成情報(以下、機器構成情報)はこれらの設計情報に含まれる。OT側の情報を正しくIT側に伝達するためには、設計情報を一次情報としてIT側のデータモデルを作成するべきであるが、技術や手法の違いから情報の断絶が発生している。

2.2 データ形式の不統一 データ収集をプログラムで行う場合、収集されるデータの形式や付加されるメタ情報はプログラムの実装者に依存して差異が生じる。差異の例としてはデータの項目名や値の型・精度、単位の表記などがある。

同一の業界や組織内であれば規格や規約によって定型化を図ることは可能である。しかし、規格や規約は業界や組織ごとに異なっている。それらを遵守するかどうか実装者次第であり徹底することは容易ではない。IoTでは業界

や組織を横断したデータ収集が行われる場合も多く⁵⁾、形式の定型化はより一層困難となる。

2.3 プログラミングコスト PLCやIPCはプログラム次第で様々な処理を行うことで、ユーザの多様なニーズに対応ができるが、その反面、データ収集のように単純な処理であってもプログラムが必要になる。プログラム作成にあたってはPLCやIPCに接続するインタフェースやセンサの仕様情報をデータシート等から読み取り、パラメータとして組込むことになる。

このようなデータ収集・変換プログラムは類似性が強く、熟練のエンジニアであれば、ある程度ルーチン化できるものの、多数の機器を対象にする場合のコストは無視できない。また、練度の低いエンジニアが作成する場合、性能や品質に問題が生じるリスクが高くなる。

3. 技術内容

3.1 概要 制御機器の仕様情報を登録した「機器情報ライブラリ」と、機器情報ライブラリを参照して取得した情報からデータ収集・変換用のPLCプログラムを自動生成するソフトウェア「フォーマットングツール」を開発した。「機器情報ライブラリ」が登録対象とする制御機器はセンサ類や、それらをPLCに接続するための入出力ユニットや通信ユニットなどがある。

生成するプログラムの言語には、テキストベースであるため自動生成技術との親和性が高く、変換処理に必要な論理演算・数値計算を得意とする⁶⁾ことから国際標準規格IEC61131-3の構造化テキスト(ST言語)を採用した。

フォーマットングツールによって生成されたプログラムは、PLCに接続された機器からデータを収集し、「工業量変数」と名付けた構造体に格納する。機器から取得されたデータは「デバイス変数」と呼ばれるPLCのメモリ領域に格納され、その値は工業量そのものではなくAD変換値のような代用値であることが多い。このような場合は生成されたプログラム内で工業値に変換したうえで工業量変数に格納する。工業量変数はネットワーク変数としてPLCの外部からの読み出しや、PLCのデータベース接続機能により外部データベースへ送信することも可能である。

図1にシステム構成を示し、続いてプログラム生成からデータ収集までの流れを説明する。

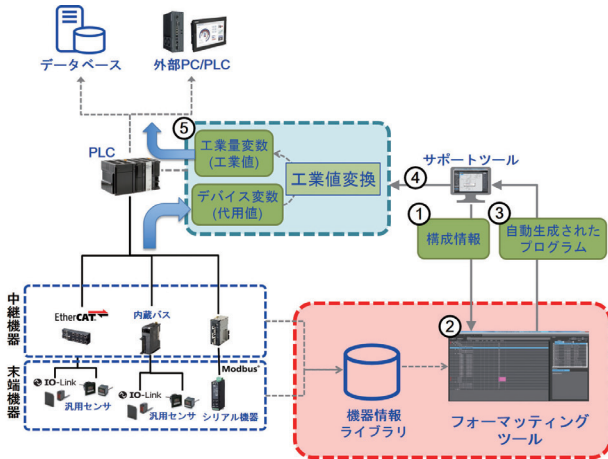


図1 システム構成

- ① PLCのサポートツール（以下、サポートツール）で設定済の中継機器（3.2に記載）までの機器構成情報をフォーマットツールに取り込む
- ② ユーザ選択操作に従い機器情報ライブラリから制御機器の工業量情報、変換情報、単位情報などを①に追加し、デバイスマップ（3.2に記載）を完成させる
- ③ 完成したデバイスマップからプログラムを自動生成する
- ④ 自動生成したプログラムをサポートツール経由でPLCに転送する
- ⑤ 転送したプログラムをPLC上で動作させて機器からのデータ収集とネットワーク公開・データベース送信を行う

3.2 デバイスマップの作成 サポートツールはPLCに接続している入出力ユニットや通信ユニット（以下、中継機器）を機器構成情報として管理している。フォーマットツールはPLCの機器構成情報を取り込み、それを中継機器の先にあるセンサなどの末端機器まで入力および設定することでデバイスマップを作成する。図2はデバイスマップの概念図である。

デバイスマップはPLCを始点、各機器をノードとするツリー構造によって定義されており、各機器は1つ以上の入力と、1つ以上の出力を持つように構造化している。この構造は接続インタフェースに依存しないものであり、接続インタフェースの違いは各入出力の属性として機器情報ライブラリに登録されている。フォーマットツールは中継機器の入力と末端機器の出力それぞれのインタフェースが一致するようデバイスマップを構築する。

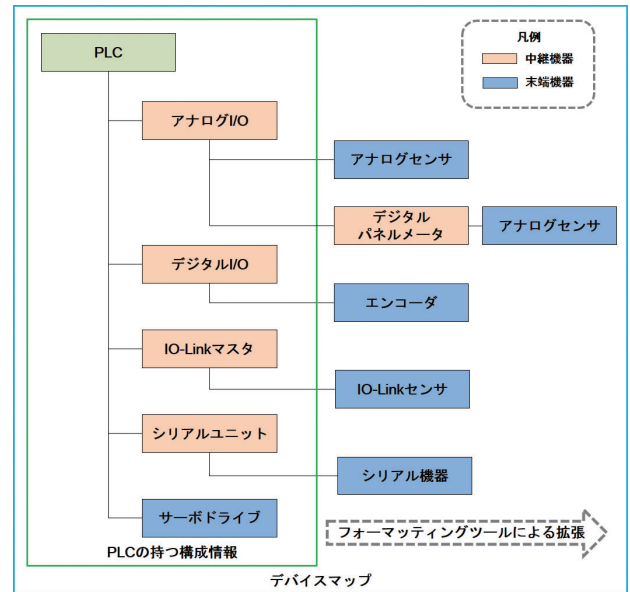


図2 デバイスマップの概念図

機器情報ライブラリを簡略化した図3と、フォーマットツールの画面（図4、図5）で具体例を説明する。サポートツールの持つ機器構成情報から中継機器の形式（①）を条件としてライブラリを検索し中継機器の入力を持つアナログなどのインタフェースをユーザに提示する（②）。提示されたインタフェースをユーザが選択したら、仕様の合致するインタフェースと工業量（③）を出力として持つ機器を列挙し、選択候補としてユーザに提示する。選択候補のいずれかをユーザが選択（④）すると、その機器のパラメータを取得して計測対象の工業量を特定する（⑤）。

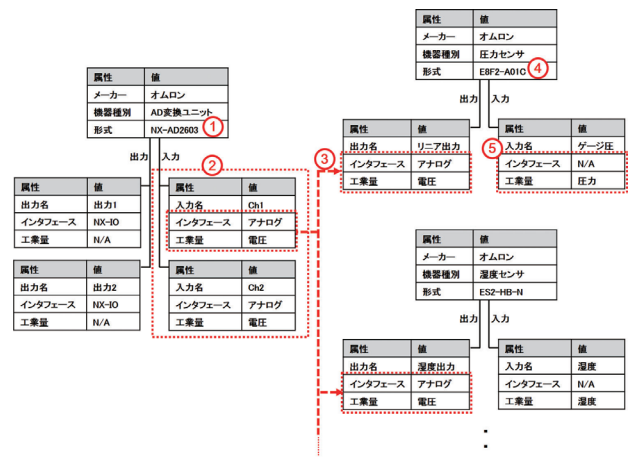


図3 機器情報ライブラリを使った機器選択



図4 選択可能機器の提示

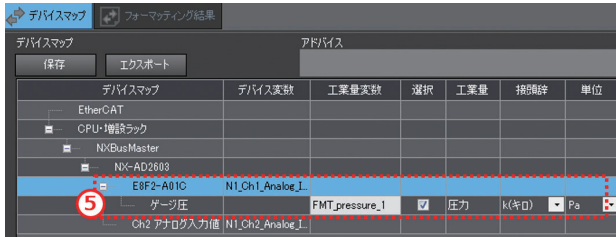


図5 工業量の特定

3.3 共通雛形プログラムと機器個別パラメータへの分解

個別の機器に応じたデータ収集・変換プログラムを生成するために、接続インタフェースごとに共通の雛形プログラムと、機器個別のパラメータに分解して機器情報ライブラリに登録した。

以下、アナログセンサを例として説明する。アナログセンサは計測した工業量を電圧もしくは電流に変換して出力する。AD変換ユニットはアナログセンサの出力をデジタル値に変換する。PLCのプログラムではこのデジタル値をAD変換ユニットの仕様に従って電圧もしくは電流の値に変換したのち、アナログセンサの仕様に従って更に変換を行うことで工業量を得る。図6に示す例では計測値 50kPa が電圧値 3V を経て変換値1200としてAD変換ユニットのデバイス変数に格納される。

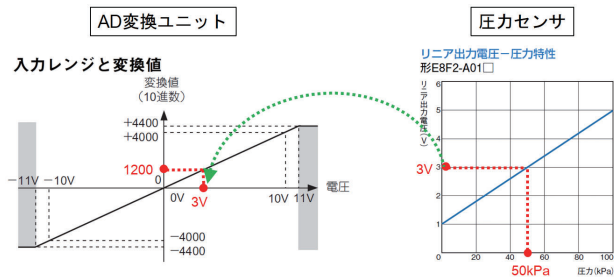


図6 計測値の取り込み

デバイス変数の値を工業量である圧力 (単位: kPa) に変換するために図7に示す線形変換式 (1) (2) を使う。各変数の意味は表1の通り。Pressureが工業量変数である。

Voltage :=
 $(\text{Analog_Input_Value} + 4000) / 8000 * 20 - 10; \dots(1)$

Pressure :=
 $(\text{Voltage} - 1) / 4 * 100; \dots(2)$

図7 工業量への変換

表1 変換式で用いる変数

| 変数名 | 内容 | 単位 | 値の例 |
|--------------------|--------|-----|------|
| Analog_Input_Value | デバイス変数 | なし | 1200 |
| Voltage | 電圧値 | V | 3 |
| Pressure | 圧力値 | kPa | 50 |

この線形変換式はアナログセンサやAD変換ユニットに共通するものであるが、傾きやオフセットを決めるパラメ

ータは個別の機器によって異なる。そこで機器情報ライブラリでは (1) (2) のような変換式を生成するために図8で示す共通の雛形プログラムを登録している。雛形プログラムには $\{ \}$ で囲まれたパラメータが組み込まれている。線形変換式でのパラメータ一覧を表2に示す。

$$\begin{aligned} \{\text{output}\} := & \\ & (\{\text{input}\} - \{\text{outMin}\}) / (\{\text{outMax}\} - \{\text{outMin}\}) * \\ & (\{\text{inMax}\} - \{\text{inMin}\}) + \{\text{inMin}\}; \end{aligned}$$

図8 線形変換式の雛形プログラム

表2 線形変換式のパラメータ一覧

| パラメータ名 | 内容 | 意味 |
|--------|-----|-------|
| input | 変数名 | 変換前の値 |
| output | 変数名 | 変換後の値 |
| inMax | 数値 | 入力上限 |
| inMin | 数値 | 入力下限 |
| outMax | 数値 | 出力上限 |
| outMin | 数値 | 出力下限 |

機器情報ライブラリ内には機器ごとにその仕様に基づくパラメータの具体的な値を登録している。フォーマットツールは雛形プログラム内のパラメータをこの具体的な値で置き換えて変換式を自動生成する。図8の場合、AD変換ユニットでは表3の値に置き換えることで図7の (1) 式が生成され、圧力センサでは表4の値に置き換えることで図7の (2) 式が生成される。

表3 AD変換ユニットのパラメータ例

| パラメータ名 | 値 |
|--------|--------------------|
| input | Analog_Input_Value |
| output | Voltage |
| inMax | 10 |
| inMin | -10 |
| outMax | 4000 |
| outMin | -4000 |

表4 圧力センサのパラメータ例

| パラメータ名 | 値 |
|--------|----------|
| input | Voltage |
| output | Pressure |
| inMax | 0 |
| inMin | 100 |
| outMax | 1 |
| outMin | 5 |

上記の例は単純なパラメータ置換であるが、雛形プログラムからのプログラム生成ではパラメータの値に沿って分岐や繰り返し処理を使うことも可能である。これにより符号の有無によるキャスト処理や、エンコーダが出力するピッ

ト列をデコードし測定値を抽出する処理などを共通の雛形プログラムから生成できる。また、変数値の変換処理だけでなく、通信プログラムであっても生成できる。例えば Modbus RTUであればコマンド送受信処理を雛形化し、そこに機器情報ライブラリに登録したファンクションコードやレジスタアドレスなどの機器個別パラメータを埋め込むことでシリアル通信プログラムを自動生成できる。

表5に示す5つの接続インタフェースを例に共通処理の雛形プログラム化とパラメータへの分解を行い、個別の機器に対応するプログラムの生成ができることを確認した。

表5 接続インタフェースと共通処理

| インタフェース | 共通処理 | 主なパラメータ |
|------------|----------|--------------------------------|
| アナログ | 線形変換 | 入出力の上下限 |
| エンコーダ | ビットの結合 | ビット数 |
| | デコード | エンコード方式 (BCD、グレイコード) |
| IO-Link | ビット演算 | ビットオフセット ビット長 |
| | キャスト | 符号の有無 |
| | 線形変換 | 傾きとオフセット |
| EtherCAT | 線形変換 | 傾きとオフセット |
| Modbus RTU | コマンド送受信 | ファンクションコード レジスタアドレス ワード数 |
| | エンディアン変換 | エンディアン種別 |
| | キャスト | 符号の有無 |
| | 線形変換 | 傾きとオフセット |

3.4 設定選択によるパラメータ切り替え 変換式に組み込まれるパラメータが設定によって変化する機器に対応するため、機器情報ライブラリは設定ごとにパラメータを変更可能なように設計されている。例えば入力仕様が切り替えられるAD変換ユニットの場合、表6に示すようなパラメータ群を機器情報ライブラリに登録する。

表6 設定によってパラメータが変化する例

| 設定名 | 工業量 | inMax | inMin | outMax | outMin |
|---------|-----|-------|-------|--------|--------|
| -10-10V | 電圧 | 10 | -10 | 4000 | -4000 |
| 0-5V | | 5 | 0 | 8000 | 0 |
| 1-5V | | 5 | 1 | 8000 | 0 |
| 0-10V | | 10 | 0 | 8000 | 0 |
| 4-20mA | 電流 | 20 | 4 | 8000 | 0 |

図9のようにフォーマットツールを通して設定名を選択することで、適切なパラメータが適用される。

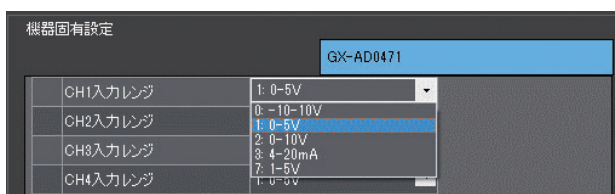


図9 設定選択によるパラメータ決定

3.5 データ形式の定型化と外部への送信 工業量変数の格納先として表7に示すPLC内の構造体を自動生成する。

表7 収集データの構造体定義

| 項目 | データ型 | 内容 |
|----------------------------|---------------|------|
| TimeStamp | DATE_AND_TIME | 収集時刻 |
| FMT_<識別子> | LREAL | 工業値 |
| FMT_<識別子>_unit | STRING | 単位 |
| FMT_<識別子>_physicalQuantity | STRING | 工業量名 |

収集したデータのメタ情報である単位および工業量は機器情報ライブラリ内にあるリストから選ばれ、文字列型として格納される。このリストはSI国際文書および日本の計量法、計量単位令で定められた表記に従って作られており、属人的な表記の差異が生じないようにしている。

同じくメタ情報である収集時刻にはPLCの内部時計の時刻を使うことで、すべてのデータの収集時刻が同一の基準で記録されるようにした。

PLCがデータベース接続機能を搭載している場合、フォーマットツールは工業量変数の値をPLCから上位システムにあるデータベースのテーブルに格納するプログラムを生成する。またフォーマットツールは、上記の格納のための、データベースのテーブルを定義するSQL文も生成する。このSQL文をデータベースに対し実行することでPLCとデータベースのデータ構造を同期して構築できる。(図10参照)

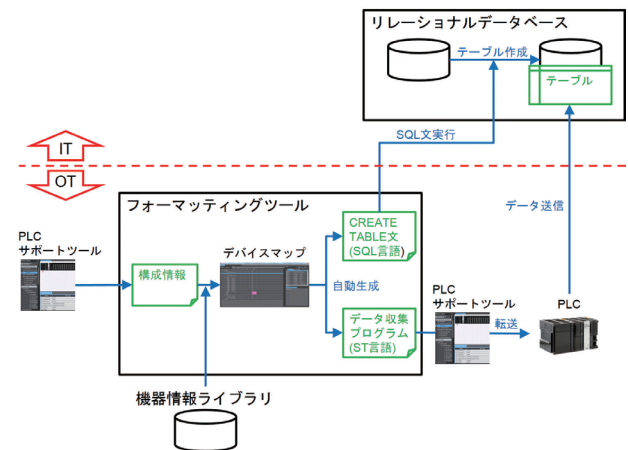


図10 データベース送信のデータフロー

4. 効果

機器情報ライブラリは制御機器仕様というOTドメインの知識の一種をライブラリ化したものと言える。また、サポートツールの持つ機器構成情報は装置の設計情報の一部である。フォーマットツールはこれらを一次情報として、工業量変換した情報を表現する定型の構造体と、それを収集・変換・送信するプログラムを自動生成する。またIT層のデータモデルとしてデータベースのテーブル定義を自動生成する。

すなわちOTおよびITの異なる2つの分野の知識・スキルをツール化することで同一のエンジニアによる作業で完結したデータベースへの送信の実行が可能になった。これにより2.1で述べたコミュニケーションでの遅れや齟齬を回避し、OT-IT間の情報断絶を軽減することが期待できる。

フォーマットツールは機器情報ライブラリの内容に従って一定の規則のもとに構造体の定義を自動生成するため、同じ機器構成に対して同じ結果が得られる。これにより2.2で述べたような作業への依存なしにデータ形式を統一することができる。

データの定型化には本稿の技術の他に、ゲートウェイデバイスと呼ばれるデータハブ機能を担う専用装置を用いる方法や、クラウドを含むITシステム上で変換処理を行う方法がある。これらに対する本稿での取り組みの利点としては、PLC自身がデータ収集を担うため、生産タクトや制御周期に同期したデータを扱いやすいことや、装置の設計情報を活用できることなどが挙げられる。

フォーマットツールは機器情報ライブラリを利用することでデバイスマップ作成時にユーザが機器および設定の選択という簡単な操作をするだけで、プログラムの生成に必要な情報を特定することができる。

このため2.3で述べたように従来の方法で必要とされたデータシートからの仕様情報の読み取りが不要となる。また、仕様の読み誤りによるプログラミングミスを抑止し、デバッグ工数を低減できる。これにより、データ収集開始までの費用・期間増の要因となるプログラム作成コストの低減を図ることができる。

5. むすび

本稿での取り組みにより自動生成されたプログラムを使ってPLCをデータハブとすることで、アナログセンサを含めた多様な制御機器からの生データを体系的にIoTデータに変換するというアイデアの実現性と有用性を実証できた。

現在の機器情報ライブラリに登録されている機器は限られたものである。今後市場に存在する多くの機器をカバーするためにはパラメータの登録や更新を管理・運営する仕組み作りが求められる。

また、今回はデータの送信先をリレーショナルデータベースとしたが、これを各種のIT接続に展開することで、FA現場の情報を多様なITエコシステムと結合するシステムを構築する際の一助となることを目指したい。

参考文献

- 1) 藤本隆弘. 能力開発競争. 中央公論新社, 2003, 406p.,
- 2) 総務省. 厚生労働省. 文部科学省. 2018年版ものづくり白書. 2018, 325p.,
- 3) オムロン株式会社. 共創型現場データサービス i-BELT. オムロン制御機器. 2018-09-20. <https://www.fa.omron.co.jp/solution/i-belt/>. (参照 2018-12-19) .
- 4) Industrie 4.0 Platform. インダストリー4.0実現戦略, 2015.,
- 5) スマートマニュファクチャリング特別委員会. 2016年度版 製造業2030. (一社) 日本電機工業会, 2016, 76p.,
- 6) (社)日本配電制御システム工業会ほか. はじめてのIEC61131-3. PLCopen Japan, 2011, 105p.,

執筆者紹介



小林 義明 Yoshiaki Kobayashi
 インダストリアルオートメーションビジネス
 カンパニー
 技術開発本部
 第1技術部
 専門: ソフトウェア工学

本文に掲載の商品名は、各社が商標としている場合があります。