

GX Works3 における iQ-F シリーズシーケンサの パラメータ省力設定に向けた開発

名古屋事業所 FAエンジニアリングソフトウェア統括部 第一技術部
エンジニアリングソフトウェア開発第三課

・平良 賢一
・大石 修司
・林 晃良

要 旨

MELSOFT GX Works3 (以下“GX Works3”という) は国際規格 IEC61131-3 準拠の高水準言語によるプログラミング環境とコンパイラ・デバッガの機能を備え、“書くから選ぶ”をコンセプトに操作性及び部品化・構造化を強化した三菱電機(株)のシーケンサエンジニアリングソフトウェアである。

シーケンサは工場自動化 (Factory Automation) を実現する制御機器であり、GX Works3 が対応するシーケンサには三菱電機(株)の MELSEC iQ-R シリーズおよび MELSEC iQ-F シリーズ (以下“iQ-F シリーズ”という) がある。本稿で述べる iQ-F シリーズは、Ethernet やアナログ、SD メモリカードスロットなど多彩な内蔵機能を備えた小規模・スタンドアロン向けマイクロシーケンサである。また、iQ-F シリーズは“IoT に使える機能”として製造ライン同士の情報を共有するシンプル CPU 通信機能、離れた場所からアクセスする FTP サーバ機能や Web サーバ機能、トラブル対策としてデータロギング機能やリアルタイムモニタ機能なども有しており、製造ラインの IoT 化の第一歩を実現するこ

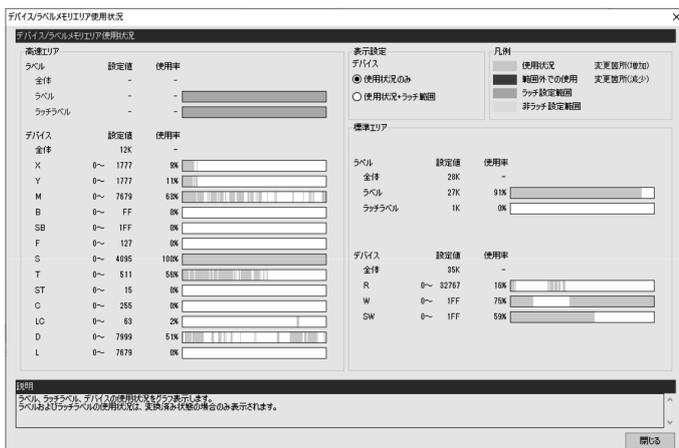
とができる製品である。

GX Works3 では一般的なプログラミングの変数にあたるラベルを使うことができ、プログラムの可読性を高めることができる。

ラベルはシーケンサのメモリエリア内のラベルエリアに格納される。iQ-F シリーズは小規模向けに低コストを実現した製品であり、他のシリーズと比べメモリエリアのサイズが小さい。このため、確保できるラベルエリアも小さく、ラベル使用量増加に伴いラベルエリアの空き容量が不足する頻度が高い。

ラベルエリアの空き容量が不足した場合、シーケンサのパラメータを変更することでラベルエリアを増やすことができる。ラベルエリアを増やすには現在のラベル使用量などの情報を収集した上で、複数のパラメータ変更が必要であり、手順が煩雑という課題があった。

本稿では、ラベルエリアを増やす場合の GX Works3 iQ-F シリーズメモリエリアパラメータ省力設定の開発で取り組んだ技術、工夫した点について述べる。



GX Works3 ラベルメモリエリアパラメータの省力設定図

1. まえがき

MELSOFT GX Works3 (以下“GX Works3”という) は国際規格 IEC61131-3 準拠の高水準言語によるプログラミング環境とコンパイラ・デバッガの機能を備えたシーケンサエンジニアリングソフトウェアである (図1)。

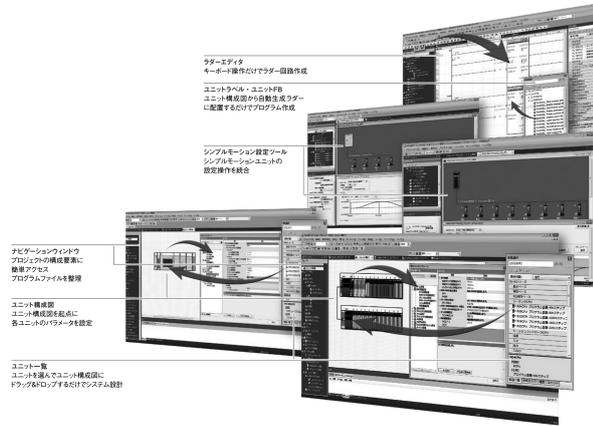


図1. IEC 準拠の主要なプログラミング言語に対応した GX Works3

(三菱電機カタログ“シーケンサエンジニアリングソフトウェア GX Works3”L (名) 08333-C より引用)

シーケンサは工場自動化 (Factory Automation) を実現する制御機器であり、GX Works3 が対応するシーケンサには MELSEC iQ-R シリーズ (以下“iQ-R シリーズ”という) および MELSEC iQ-F シリーズ (以下“iQ-F シリーズ”という) がある。iQ-R シリーズは中・大規模向けにビルディングブロックタイプの特徴を持つシーケンサであり、iQ-F シリーズは、Ethernet やアナログ、SD メモリカードスロットなど多彩な内蔵機能を備えた小規模・スタンドアロン向けマイクロシーケンサである (図2)。



図2. iQ-F シリーズ

(三菱電機カタログ“MELSEC iQ-F 総合カタログ” L (名) 08394-L より引用)

iQ-F シリーズ向けに、煩雑なパラメータ変更手順を改善する機能を開発したので、本稿で詳細を説明する。

2. ラベルを利用したプログラミングの課題

GX Works3 における制御システム開発の流れを図3に示す。本稿で述べる図3の「2) プログラミング」工程では、シーケンサの入力インターフェースおよび出力インターフェースを用いた制御システムを実現するプログラムを作成する。

シーケンサの入力インターフェースには押ボタンスイッチや切換えスイッチなどの入力機器、出力インターフェースには信号灯やデジタル表示器などの出力機器を接続する。シーケンサは入力機器の状態を入力インターフェースの X デバイスで保持し、出力機器への信号は出力インターフェースの Y デバイスを介して制御する。また、プログラムの演算結果などの数値データを格納する D デバイスなどがある。

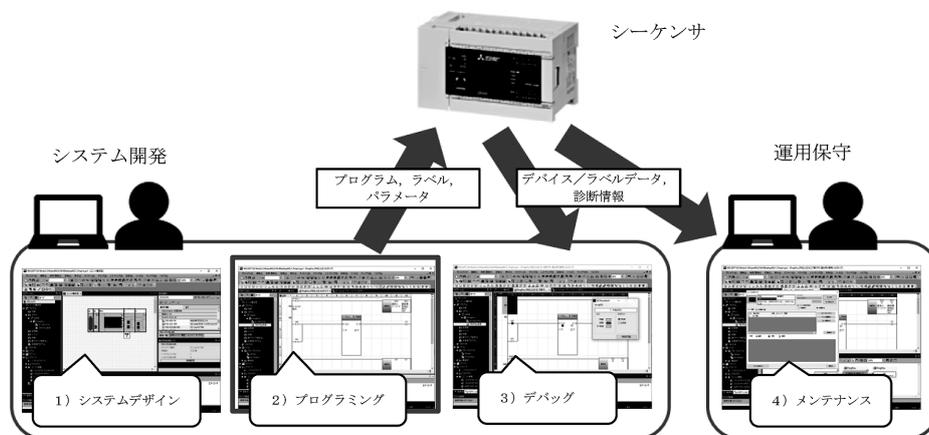


図3. GX Works3 による制御システムの開発

GX Works3ではこれらデバイスの代わりに一般的なプログラミングの変数にあたるラベルを使うこともできる。ラベルはラベルエディタにてラベル名やデータ型を指定することで、ラダー言語やST言語などのプログラムに記述することができ、プログラムの可読性を高めることができる(図4)。

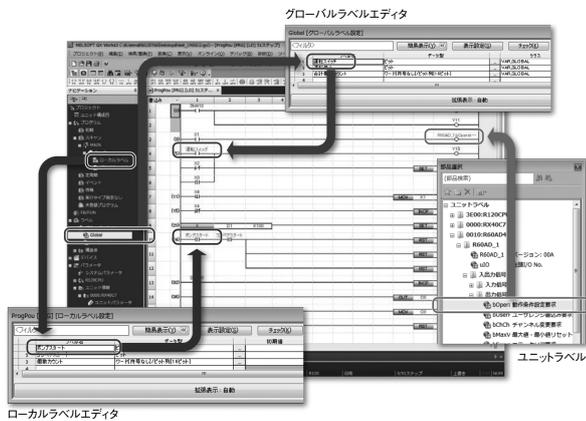


図4. ラベルを利用したプログラミング
(三菱電機カタログ「シーケンサエンジニアリングソフトウェア GX Works3™L (名) 08333-C」より引用)

シーケンサにはデータを保持するメモリエリアがある。メモリエリアは、デバイスエリアとラベルエリアで構成されており、各エリアのサイズは変更可能である(図5)。

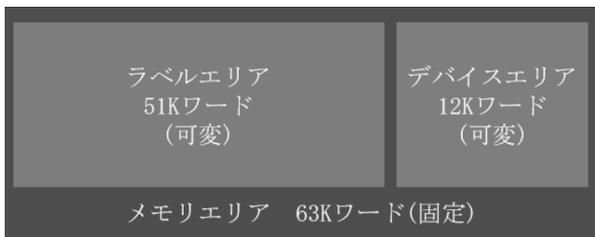


図5. iQ-Fシリーズのメモリエリアのイメージ

メモリエリア全体のサイズは、例えばiQ-FシリーズのFX5Uは63Kワード固定である。iQ-Fシリーズは小規模・スタンドアロン向けに低コストを実現した製品であり、他のシリーズと比較してメモリエリア全体のサイズが小さい。このため確保できるラベルエリアも小さく、ラベルを多く使うとラベルエリアの空き容量が不足してくる。ラベルエリアの空き容量が不足した場合、ユーザはデバイス、ラベルの使用量を収集した上で、シーケンサの各機能を設定するCPUパラメータに含まれるメモリエリアパラメータを変更し、ラベルエリアを増やす必要がある。このラベルエリアを増やす手順が煩雑である課題があった。本課題を解決するため、メモリエリアパラメータ省力設定の開発を行った。

3. メモリエリアパラメータ省力設定の開発

ラベルエリアを増やす場合、デバイスエリアを減らしてメモリエリア全体の空き容量を確保した上でラベルエリアを増やすパラメータの変更が必要となる。デバイスエリアを減らすには各デバイス点数を削減するパラメータの変更も必要である。これらラベルエリアを増やす一連のパラメータ変更手順を図6の手順1~4に示す。

図6の手順を実施することでデバイス点数を削減した分だけラベルエリアを増やすことができる。どのデバイスをどれだけ削減するかは、ユーザのシステムによって異なる。そのユースケースを表1に示す。

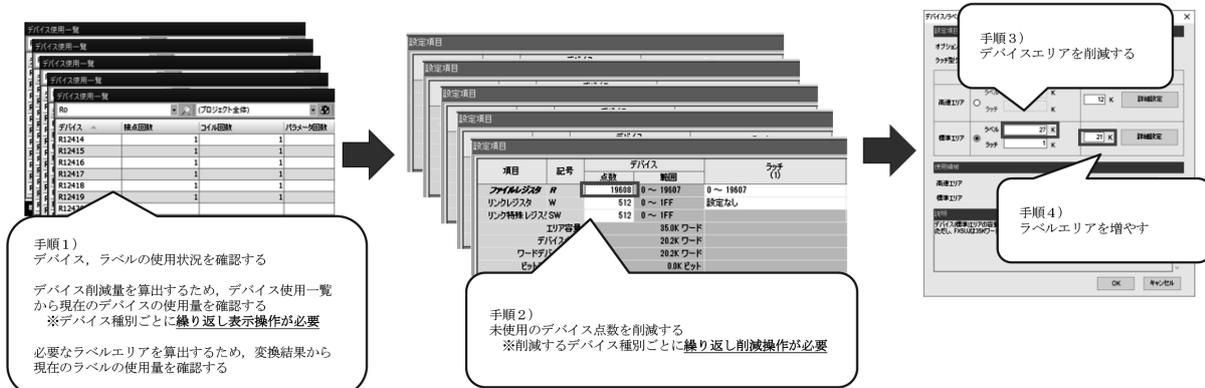


図6. ラベルエリアを増やすパラメータ変更手順

表 1. デバイス点数削減でのユースケース

| |
|---|
| ユースケース 1) 今後制御システムの機能追加を予定しており、使用予定のデバイス点数を確保した上で不要なデバイス点数を削減する |
| 例えば D デバイスは現在未使用だが今後使用予定のため 8,000 点分確保し、他の未使用デバイスはすべて削減するなど、削減対象デバイスや削減量を個々に判断して削減する。 |
| ユースケース 2) 現在のデバイスの使用状況に合わせて不要なデバイス点数を削減する |
| 削減対象デバイスや削減量を個々に判断せず、未使用のデバイスから優先して削減する。 |

表 1 のユースケース 1 において、手順 1 の「デバイス、ラベルの使用状況確認」で課題があり対策を実施した (表 2)。

表 2. 手順 1 の使用状況確認における課題と対策

| |
|---|
| 課題) パラメータ変更に必要な情報の収集が煩雑 |
| パラメータを変更する際、現在のデバイスの使用量やラベルの使用量を把握する必要がある。これら情報を収集するためには、GX Works3 の「デバイス使用一覧」でデバイス使用量を確認する、プログラム変換結果からラベル使用量を確認するなど、個々に確認する必要があった。 |
| 対策) デバイス/ラベル使用状況一覧画面を開発 |
| 画面上のボタンワンクリックでデバイスやラベルの使用状況を一覧表示する機能を開発した。これにより、パラメータ変更に必要な情報収集の手間が大幅に削減可能となった。(詳細は 3.1 節を参照) |

表 1 のユースケース 2 では、手順 1 ~ 4 全体で課題があり対策を実施した (表 3)。

表 3. 手順 1 ~ 4 のパラメータ変更における課題と対策

| |
|---|
| 課題) パラメータの変更手順が煩雑 |
| 削減対象デバイスや削減量を個々に判断する必要がないにも関わらず、手順 1 ~ 2 をデバイスごとに繰り返し実施した上で、手順 3 ~ 4 を実施する必要があり、煩雑であった。 |
| 対策) メモリエリア一括設定機能を開発 |
| 画面上のボタンワンクリックで手順 1 ~ 4 を自動で実行する機能を開発した。これにより、ラベルエリアを増やす手間が大幅に削減可能となった。(詳細は 3.2 節を参照) |

3.1 デバイス/ラベルメモリエリア使用状況一覧機能

デバイス、ラベルの使用状況が確認できる機能を開発した (図 7)。

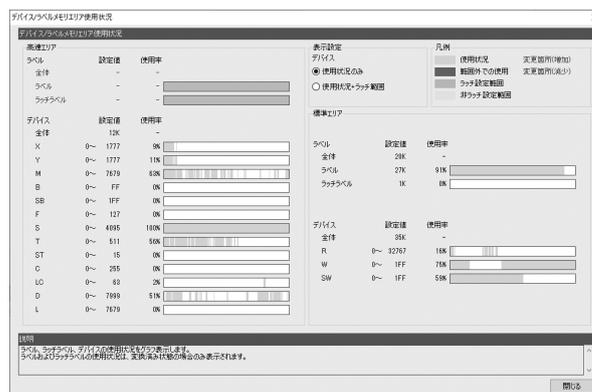


図 7. デバイス/ラベルメモリエリア使用状況画面

本機能は今後拡充が見込まれる機能であったため、保守性や拡張性を確保することを目的に、表 4 で示す非機能要件とした。

表 4. デバイスラベルメモリエリア使用状況機能の非機能要件

| |
|--|
| 要件 1) デバイス種別の増減に耐えうる設計とすること |
| シーケンサのシリーズによってデバイス種別が異なるため、別シリーズのシーケンサにも対応する必要がある。画面に表示するデバイス数が異なっても、容易に表示を変更できる設計とすること。 |
| 要件 2) 画面サイズの変更に耐えうる設計とすること |
| 今後、表示するデバイス数が増えた場合、画面サイズも大きくなるため、画面サイズを変更する必要が想定される。このため、将来、画面サイズを容易に変更できる設計とすること。 |

3.1.1 表示対象の増減を容易に実現するソフトウェア設計

今後デバイス種別の増減に耐えうる設計とするために次の (1)、(2) の手法を取り入れ、保守性や拡張性を向上させた。

(1) View コントロールの部品化・階層化

UserControl を利用することで、独自のカスタムコントロールを作成した。今後対応するデバイスが増えた場合でも容易に追加設計できるように、デバイス単位で簡単に使用できる表示部品を用意した。

加えて、デバイス/ラベルメモリエリア使用状況画面の構成を以下の階層構造として設計した (図 8, 図 9)。

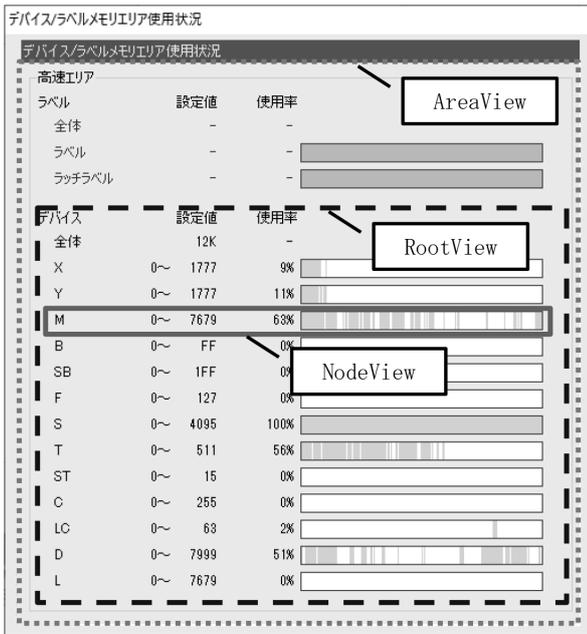


図8. View コントロールの種類

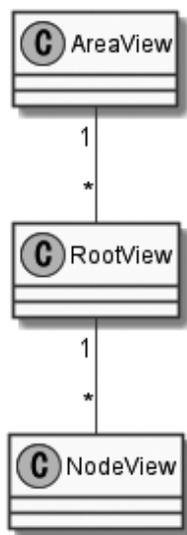


図9. View コントロールの階層構造

各 View を階層構造にすることにより、今後表示するデバイスやエリアが増えた場合でも、まとめて追加する場合と単一で追加する場合のそれぞれで柔軟に追加できるようにした。

(2) Factory Method パターンによる

アーキテクチャ設計

今後表示するデバイス種別の増減が想定されるため、この拡張性を確保するために Factory Method パターンを採用した (図 10)。

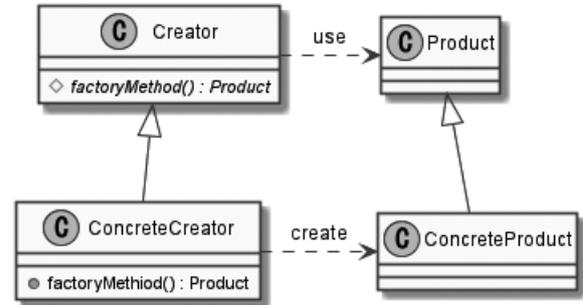


図 10. Factory Method パターン概要

表示データについて、データオブジェクトの生成手順のみ定義する Creator クラスとデータ自体の Product クラスに分離した。データ生成の処理とデータ自体が持つ処理を明確に分けることで、今後表示対象データが増えた場合でも処理を複雑化せず容易に追加設計できるアーキテクチャとした。

3. 1. 2 グラフのサイズ変更を容易に実現する

ソフトウェア設計

今後画面サイズ変更に伴いグラフの表示幅を変更することを容易にするため、次の (3) の手法を取り入れ拡張性を向上させた。

(3) グラフの表示幅に合わせた描画処理

今後画面サイズや各グラフの表示幅を任意に変更できるよう、常にグラフの表示幅に合わせて使用状況を描画するよう設計した。

ラベルのグラフは使用量を表示する (図 11)。



図 11. ラベル使用状況のグラフ表示

このため、グラフ表示幅に対して先頭から使用率分の色変更を行う処理にて実現した。

一方、デバイスのグラフは使用箇所を表示する (図 12)。

| デバイス | 設定値 | 使用率 |
|------|---------|-----|
| 全体 | 12K | - |
| X | 0~ 1777 | 9% |
| Y | 0~ 1777 | 11% |
| M | 0~ 7679 | 63% |

図 12. デバイス使用状況のグラフ表示

具体的には、設定されたデバイス点数のうち、どのアドレスを使用しているかサマリ表示する。デバイス設定点数に応じて1ピクセルあたりの表示点数を算出し、使用しているデバイスアドレスをマッピングして色変更する処理で実現した。

このため、今後画面サイズを任意に変更できる仕様になった場合、画面サイズ変更後に各グラフの再描画処理を実行するだけで変更後の画面サイズに合わせたグラフ表示が可能となった。

3.2 メモリエリア一括設定機能

メモリエリアパラメータを自動変更する一括設定機能を開発した(図13)。



図 13. デバイス/ラベルメモリエリア設定画面

本機能は画面上のボタンをクリックするだけで図5のパラメータ変更手順1~4を自動で実行するため、ラベルエリアを増やす手間を大幅に削減できる。

本機能は使用容易性向上を目的に、表5で示す機能要件とした。

表5. メモリエリア一括設定機能の機能要件

| |
|---|
| 要件) ユーザによるパラメータ見直しを必要としない一括設定であること |
| デバイスやラベルの使用状況を解析し、ユーザが期待する結果を算出するアルゴリズムであること。 |

3.2.1 ユーザによる手動設定を想定したアルゴリズム設計

一括設定機能はユーザが期待する結果とならなければならない。例えば一括設定機能によりユーザが必要とするデバイス点数が削減され、期待よりもラベルエリアが増えてしまうとパラメータの見直しが必要となる。このため、ユーザが手動で変更することを想定したアルゴリズムが必要となった。

ラベルエリアの増加量の制御については、空き容量のしきい値を指定する一括設定オプションを取り入れることで、必要以上に増加しない設計とした。例えば空き容量しきい値が「20%以上」の場合、ラベルエリアの空き容量が20%以上を満たす最小サイズまでエリアを拡大するアルゴリズムとした。

デバイス点数削減については、各デバイスの設定点数と使用点数を比較し、余剰なデバイス点数を算出する必要がある。対象デバイスは13種類あり、使用状況はそれぞれのプログラムで異なる。このため、未使用点数の多いデバイスを少量ずつ削減するアルゴリズムとした(図14)。

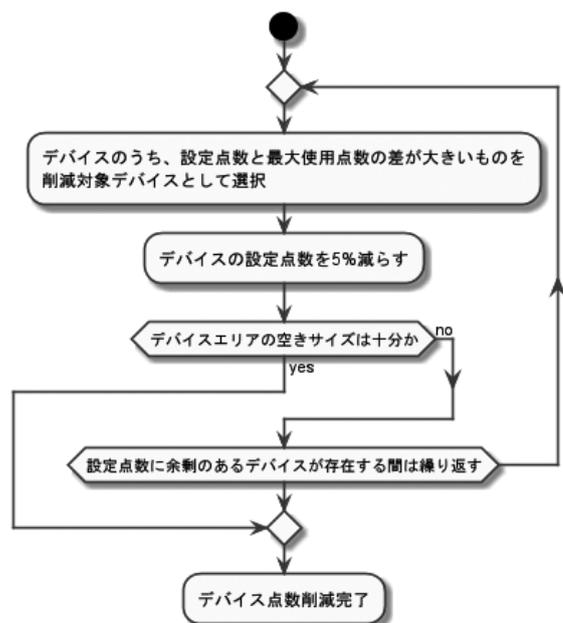


図 14. 余剰デバイス点数削減フロー

これにより、比較的余剰のあるデバイスから少しずつ削減するため、未使用デバイスがすべて削除される可能性が低い。また、未使用点数の少ないデバイスは削減されないため、パラメータ見直しの頻度を下げることができる。

デバイス設定点数はデバイス毎に異なり少ないものは16点、多いものは32,768点となる。このため、1回の削減点数を固定値にするのではなく、設定点数の5%を削減することで、未使用点数の多いデバイス群を設定点数に応じて均等に削減するアルゴリズムとした。

4. むすび

本稿では、GX Works3におけるiQ-Fシリーズシーケンサのパラメータ省力設定に向けた開発技術について述べた。GX Works3ではラベルを活用した「直感的なプログラミング環境による開発コスト削減」をコンセプトとしており、お客様の開発コスト削減に寄与する機能を開発した。今後もGX Works3のコンセプトの実現をさらに進めることでお客様の開発コスト削減に寄与していきたい。

最後に、本開発にあたり、ご指導・ご支援いただいた三菱電機(株)名古屋製作所 ソフトウェアシステム部の関係各位に深く感謝の意を表す。

商標・登録商標について

■Ethernetは、富士フイルムビジネスイノベーション株式会社の登録商標です。

■MELSEC iQ-R, MELSEC iQ-F, MELSOFT, GX Worksは、三菱電機株式会社の登録商標です。

＝ 執筆者紹介

平良 賢一

2017年入社 シーケンサの周辺ソフトウェア開発に従事。現在、名古屋事業所 FAエンジニアリングソフトウェア統括部 第一技術部 エンジニアリングソフトウェア開発第三課 グループリーダー

大石 修司

2018年入社 シーケンサの周辺ソフトウェア開発に従事。現在、名古屋事業所 FAエンジニアリングソフトウェア統括部 第一技術部 エンジニアリングソフトウェア開発第三課

林 晃良

2020年入社 シーケンサの周辺ソフトウェア開発に従事。現在、名古屋事業所 FAエンジニアリングソフトウェア統括部 第一技術部 エンジニアリングソフトウェア開発第三課
