

## IEC 61131-3 と PLCopen®

### IEC 61131-3 の特長② ～プログラムの再利用性～

PLCopen Japan 代表幹事  
松隈 隆志

### 3. 可読性の向上

日本においては LD (ラダーダイアグラム) 主体の従来ツールが普及してきました。しかしながら、使用しているユーザーは下記のような課題 (不満) を抱えています。

- ① プログラムを機能毎に区分 (モジュール化) することが難しく、第三者には解読出来ない。
- ② LD は一筆書きなので、コンカレントな開発 (分業) が出来ない。
- ③ メーカー独自の応用命令を覚えなくてはならない。(複数メーカーの習得は大変)

一方 IEC 61131-3 では、図 1 のように機能単位でプログラムを作成します。この単位のことを本規格では **POU(Program Organization Unit)** と呼んでいます。

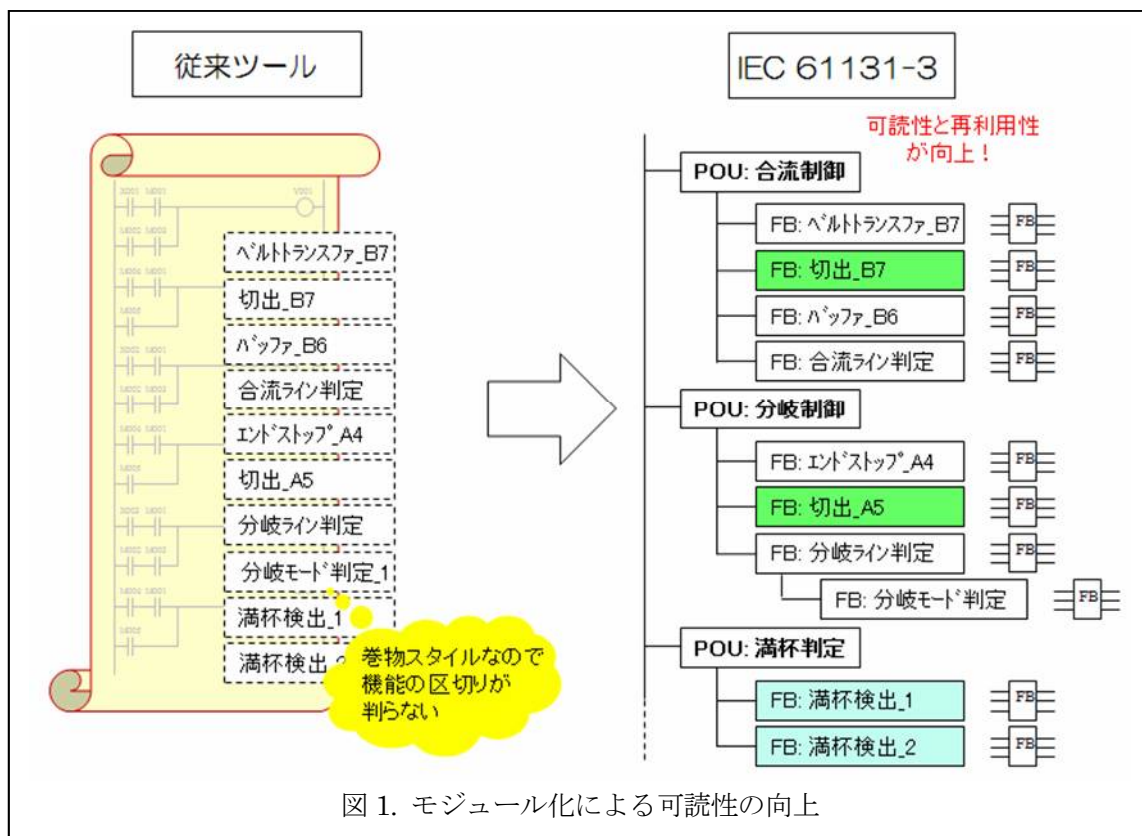


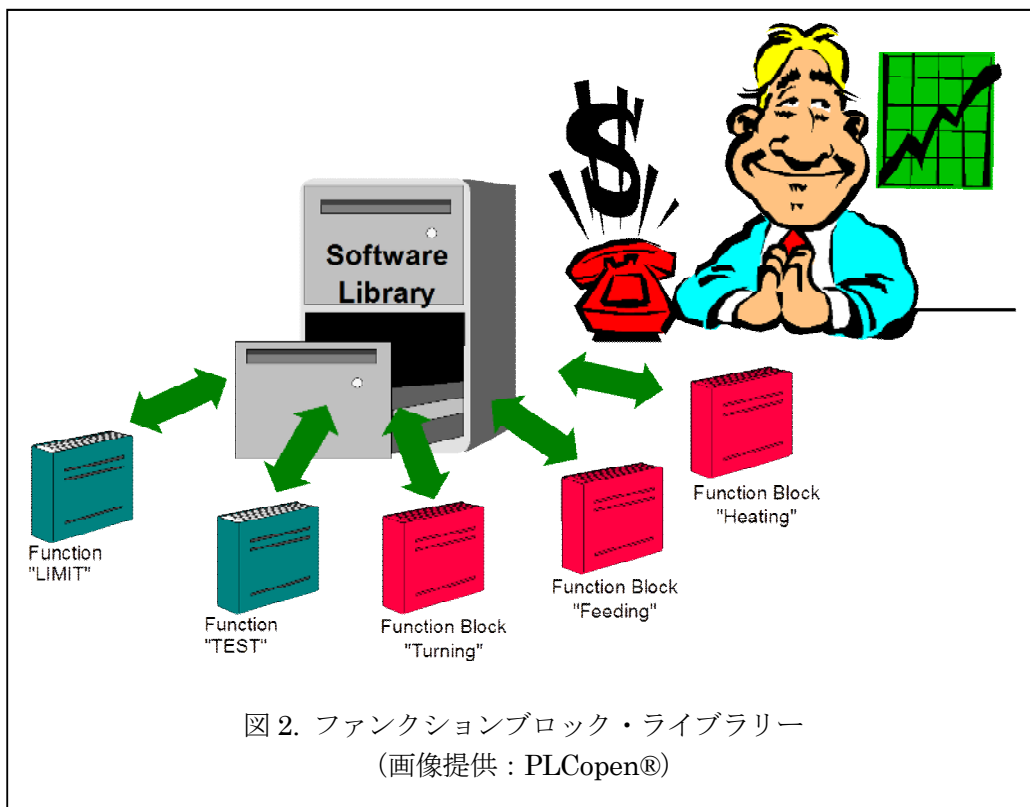
図 1 は搬送ラインをイメージした例ですが、IEC 61131-3 ではどの処理がどの POU 内に書かれているのか容易に判別ができる為、作った本人以外でもプログラムの理解・修正が可能になります。可読性の向上はデバッグやトラブルシューティングの時間を削減するだけではなく、バリデーション (設計仕様書に基づくソースコードレビュー) にも役立ちます※1)。

※1) 医薬品の製造設備・装置に使用する PLC のプログラムはバリデーションが義務付けられています。

## 4. プログラムの再利用性

IEC 61131-3 及び PLCopen®では汎用性の高い機能を標準 FB (ファンクションブロック) として規定し、PLCベンダーに実装を促しています。これらの標準 FB でカバー出来ない機能については、ユーザー自身で独自の FB を作成することも可能です。複数箇所でする機能や将来的に繰り返し使えそうなプログラムを FB として部品化し、ライブラリーファイルに登録 (保存) することでプログラム資産の再利用や共有が可能になります (図 2)。図 1 における“切出”と“満杯検出”は制御対象を区別する為にインスタンス名 (FB 変数名) を変えてありますが、同じ FB を再利用しています。実績のある FB を再利用することは、**プログラミング時間の削減**のみならず、**品質向上**にもつながります。また、FB の中に複数の FB を入れ子にすることも出来るので、構造化プログラミングにも対応しています。

昨今、グローバル化に伴うノウハウの流出が懸念されていますが、FB の中に独自アルゴリズム (計算式など) を隠蔽 (ブラックボックス化) することで、第三者の目に触れさせないことも可能です。



## 5. POU とタスク

POU には、PROGRAM、FB、FUN (ファンクション) の 3 種類が規定されています。FUN は、四則演算や型変換などに使用する内部状態を持たない命令です。つまり“入力と同じであれば演算結果は常に同じになる”というところが FB との違いです。全ての POU からアクセス可能な共通の変数のことを「グローバル変数」と呼び、その変数が定義された POU の中でのみアクセス可能な変数を「ローカル変数」と呼んでいます (図 3)。

次に、プログラムの実行単位である「タスク」について説明します。

タスク内に POU を登録することによってプログラムが実行されます。タスクには、常時繰り返し実行される「デフォルトタスク」、設定した周期毎に実行される「定周期タスク」、指定した BOOL 型変数が 0 → 1 に変化した時に 1 回だけ実行される「イベントタスク」が用意されており、POU の用途に応じて実行優先順位とタイミングを設定することができます\*2)。

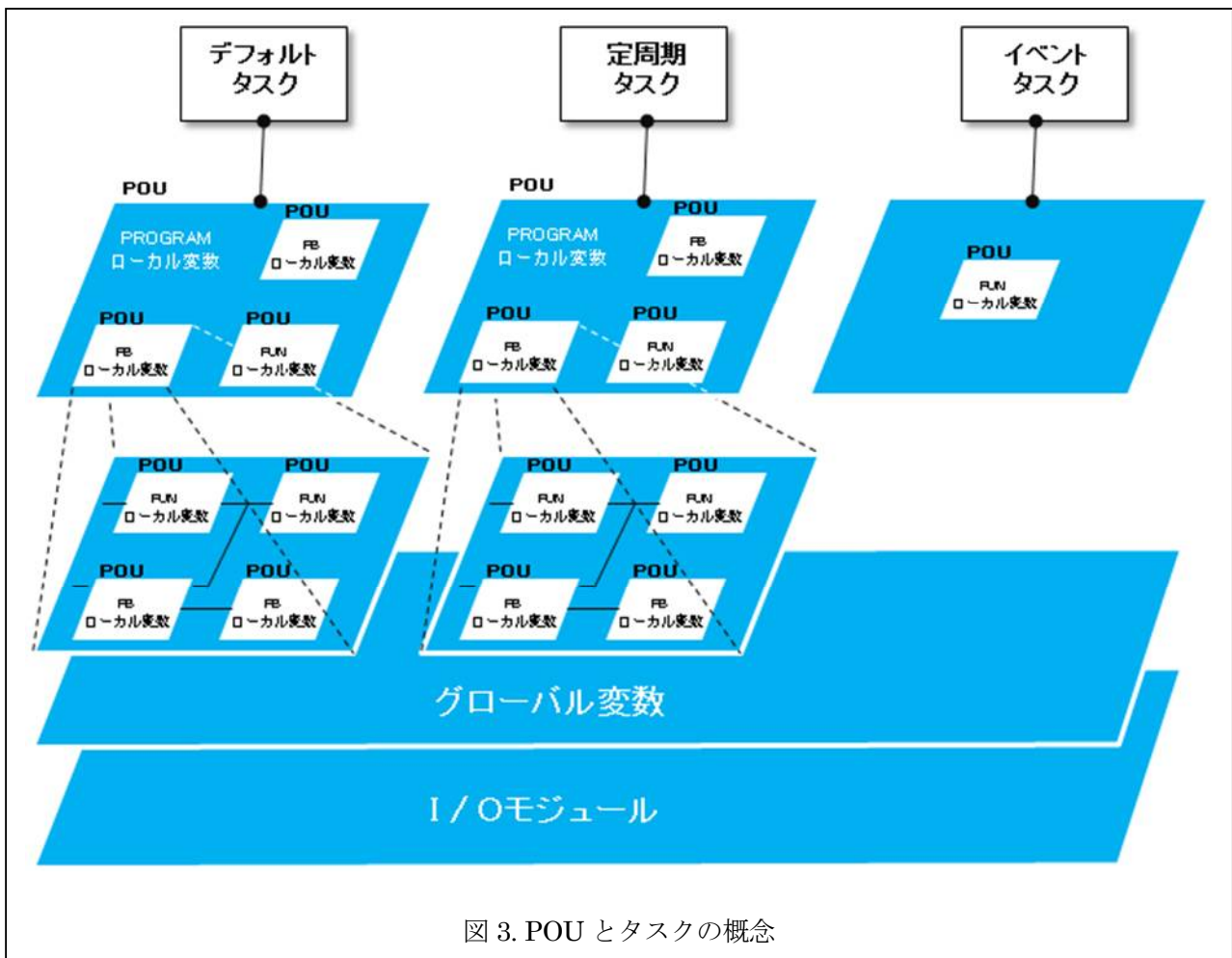


図 3. POU とタスクの概念

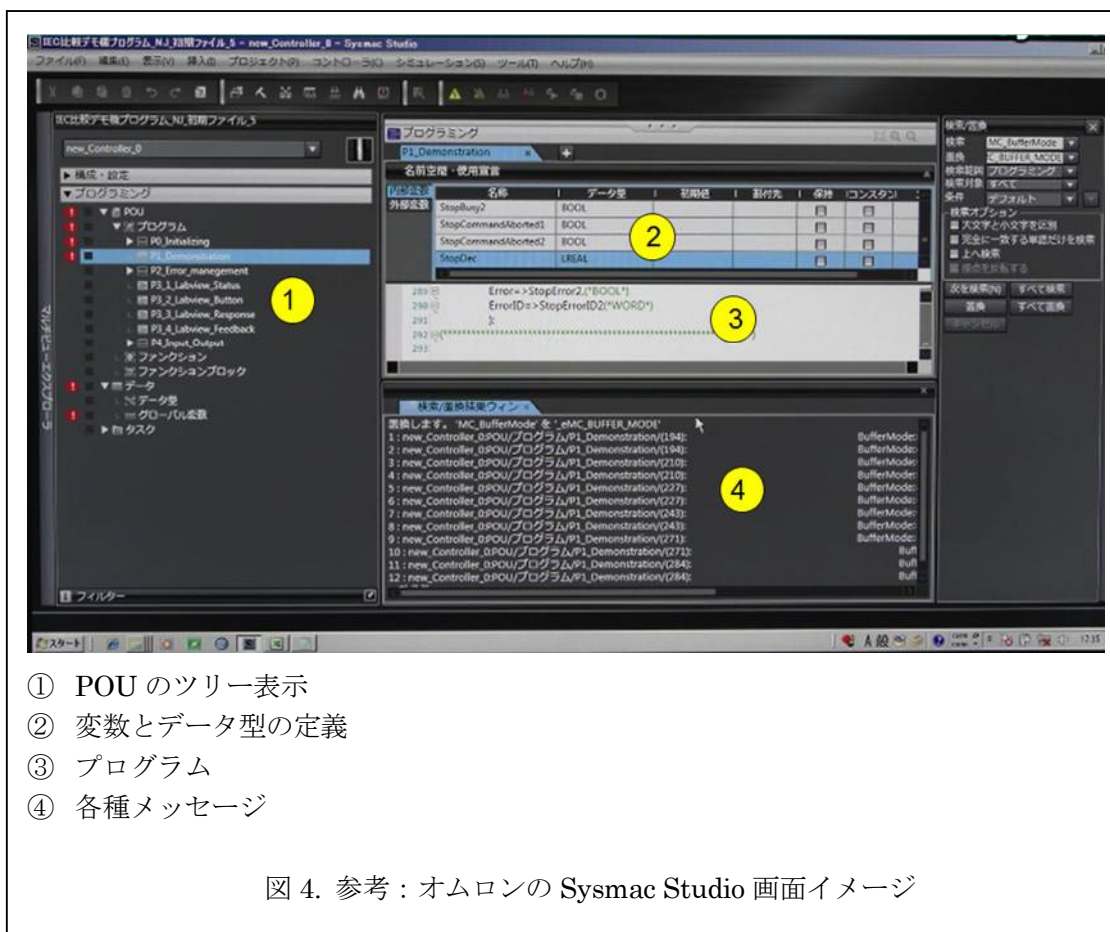
※ 2) 各タスクの呼称や最小設定単位はメーカーによって異なります。

## 6. 共通のルック&フィール

メーカー独自の従来ツールでは、メーカーが替わる毎に分厚いマニュアルを読んだり、数日間のメーカー研修を受講したりと、習得までにかかなりの時間を要してしまいます。

IEC 61131-3はプログラミング言語のみならず、プログラムの構成や変数の定義なども規定されている為、本規格に準拠したプログラミングツールの操作性はおのずと統一感がとれたものになっています。従ってA社のツールを一度習得してしまえば、B社、C社のツールも直感的に短時間で使いこなすことが出来るでしょう。

図 4は IEC 61131-3 ( JIS B 3503 ) に準拠した Sysmac Studio (オムロン製) の画面例です。



- ① POU のツリー表示
- ② 変数とデータ型の定義
- ③ プログラム
- ④ 各種メッセージ

図 4. 参考：オムロンの Sysmac Studio 画面イメージ

## 7. 万国共通言語

従来ツールに慣れ親しんだ方から「今までメーカーで決めていたアドレスを自分で決める（定義する）のは面倒だ」「ST 言語はコンピュータ言語みたいで難しい」「忙しくて勉強する余裕がない」などの声を耳にすることがあります。確かにリーマンショック頃までは IEC 61131 準拠というと欧州メーカーの PLC が殆どで、マニュアルやサービスサポートも英語が基本といった状況でした。しかし今日では、多くの国内メーカーから IEC 61131 準拠の PLC を購入することが可能になっています。勿論、日本語でのサポート体制やトレーニングコース※3)、PLCopen® Japan 執筆による解説図書なども整ってきました。

産業オートメーションに限らず、グローバル市場で“国際標準”が選ばれるのは必然と言えます。何故ならば、海外で現地技術者と共同開発する場合、あるいは海外に開発を委託する場合において、IEC 61131-3 を使った経験のある技術者は容易に見つかりますが、日本独自の従来ツールを使える技術者は海外ではなかなか見つからないからです。つまり、英語と同じように IEC 61131-3 は PLC 技術者にとって万国共通の言語なのです。

※3) PLCopen®認定トレーニングコース（2014年5月開講）

## 8. 特長のまとめ

IEC 61131-3 の主な特長について紹介してきましたが、最後にもう一度、従来ツールとの違いについて表 1 にまとめておきます。

	主な作業内容	従来ツールでの作業・機能	IEC 61131-3での作業・機能
定義作業	I/Oメモリ、変数の定義	PLCベンダー固有のメモリマップ（アドレス）毎にラベル（信号名）を決め、これが変数の定義に相当する。プログラム内で使用する補助メモリも割付が必要。	PLCベンダー固有のメモリマップは意識せず、変数を定義（名称、データ型など）。絶対アドレスは外部入出力のみ。プログラム内で使用する変数のメモリ割付は不要（ツールが自動処理）。
	変数の型を定義	用途に合わせて、メモリマップからメモリの種類で型を選択する。命令種類、データにより型を常に意識する必要がある。	変数毎にツール（エクセル表なども可）で変数の型を定義する。異なる型同士の演算を防ぎ、プログラムの信頼性が向上する。PLCベンダー・機種に非依存。
ソフト作成	応用命令の使用	PLCベンダー・機種ごとに提供された固有の応用命令を理解して使用する。	FB（ファンクションブロック）とFUN（ファンクション）がこれに相当。使用者が独自のFB、FUNを作り、ソフト部品のライブラリー化が出来る。
	機能単位のモジュール化	機能毎にサブルーチンで処理するなど。完成したプログラムは巻物状になり、作成した本人以外には理解が難しい。	POUと呼ばれる構成単位でプログラムを作成するので可読性が向上。 POU内部だけで使うローカル変数、アドレス定義不要により、ソフトの部品化、再利用性が向上。
	割り込み処理等、プログラムの処理順設定	PLCベンダー・機種ごとに提供された固有の割り込み命令、定周期実行指定命令等により実行方法を定義。	タスクの概念があり、作成したプログラム（POU）をデフォルトタスク、定周期タスク、イベントタスクに割り当てればよい。
	言語の選択	ラダー言語を主体としているが、最近ではST言語、FBD言語などを利用可能なツールもある。	IEC 61131-3の4言語1要素の中から用途に応じた選択が可能。シーケンス処理、データフロー処理、数値演算など、処理内容に応じて使い分けることが出来る。

表 1. 従来ツールと IEC 61131-3 の違い

- ・ PLCopen®および関連するロゴマークは PLCopen®が所有する登録商標です。