

スマート生産設備に向けた PLCプログラムのモジュール化検証事例

Industrial Automation Forum - Forum 2018

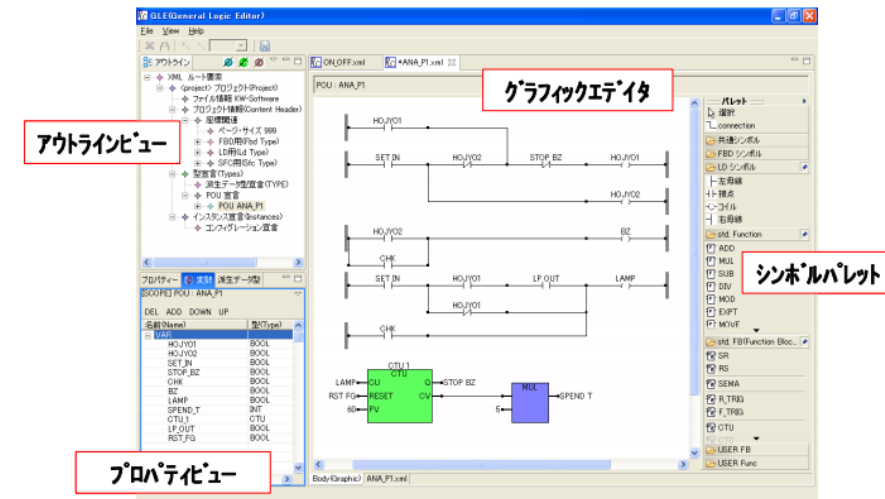
PLCopen Japan

1. PLCopen[®]とは
2. KPI連携からみたPLCopenの役割
3. PLCプログラムのモジュール化検証事例のご紹介 – SCF2017展示デモ
4. IAF 制御層情報連携（CLiC部会）における今後の活動
5. PLCopen 欧州本部の最新活動

◆ PLCopen®は、生産設備のシーケンス制御を担うPLC (プログラマブル・ロジック・コントローラ)の**プログラミング言語の国際標準規格 IEC 61131-3 の普及促進と関連技術仕様の策定**を行う非営利団体

◆ 主な取り組みは

- ✓ IEC 61131-3 仕様改訂作業への参加
- ✓ IEC 61131-3 プログラミング手法の教育・普及促進
- ✓ モーション制御FB、安全制御FB、通信FBなどの応用FBの仕様策定と認証
- ✓ 他の標準技術団体との連携、協調仕様の作成



PLCopen Japan 製

IEC 61131-3 標準プログラムエディタ

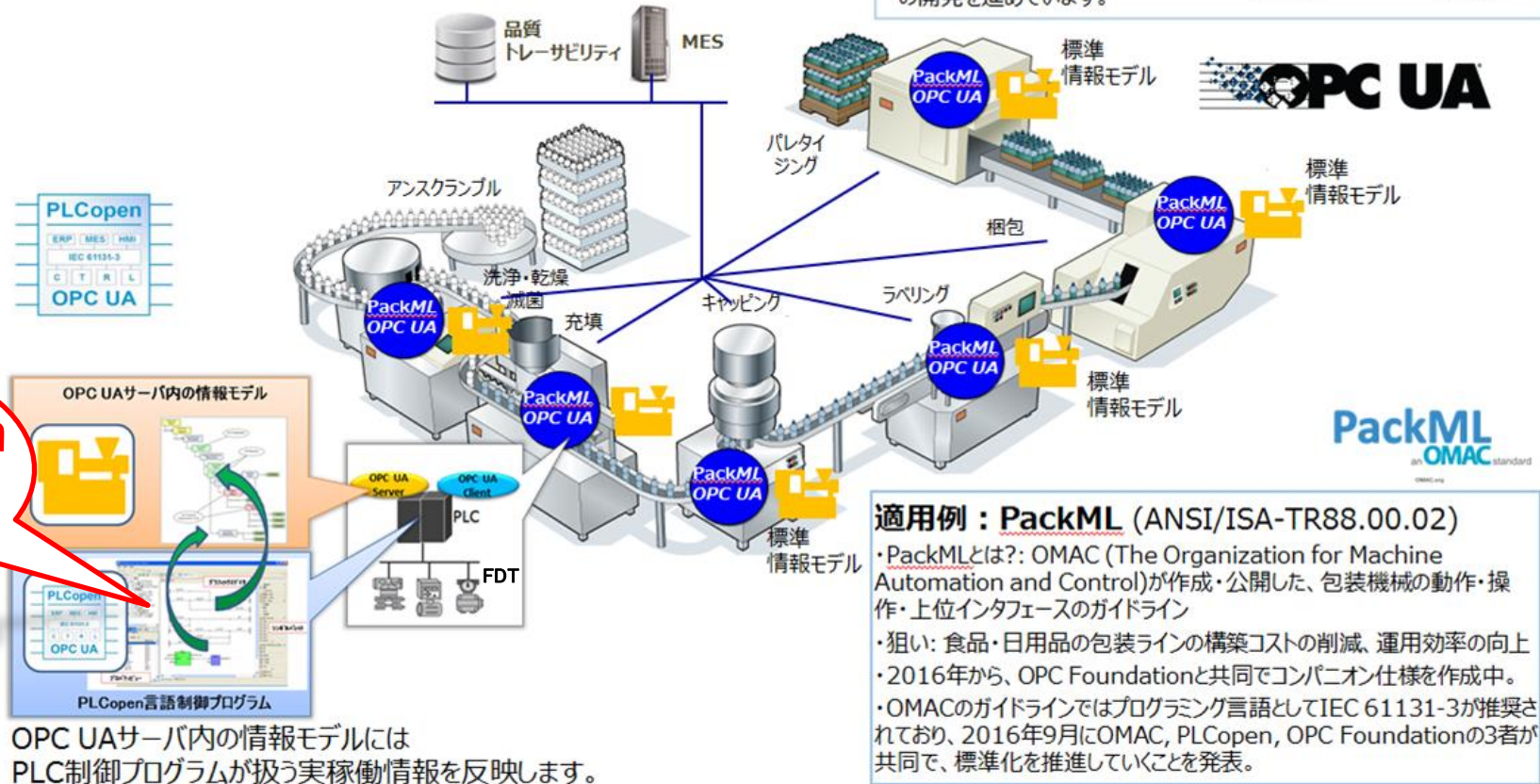
KPI連携からみたPLCopenの役割

PLCopen
for efficiency in automation

アプリケーション同士で、複雑な情報を正確かつ容易に共有できるようになり、ドメインやベンダーの垣根を越えた相互運用が実現できます。

情報モデルとは？

アプリケーション間で交換される情報の定義です。
現在、OPC Foundationは、およそ30の団体とコラボレーションし、アプリケーションに即した情報モデルの開発を進めています。



OPC UAサーバ内の情報モデルには
PLC制御プログラムが扱う実稼働情報を反映します。

適用例：PackML (ANSI/ISA-TR88.00.02)

- ・PackMLとは？: OMAC (The Organization for Machine Automation and Control)が作成・公開した、包装機械の動作・操作・上位インタフェースのガイドライン
- ・狙い: 食品・日用品の包装ラインの構築コストの削減、運用効率の向上
- ・2016年から、OPC Foundationと共同でコンパニオン仕様を作成中。
- ・OMACのガイドラインではプログラミング言語としてIEC 61131-3が推奨されており、2016年9月にOMAC, PLCopen, OPC Foundationの3者が共同で、標準化を推進していくことを発表。

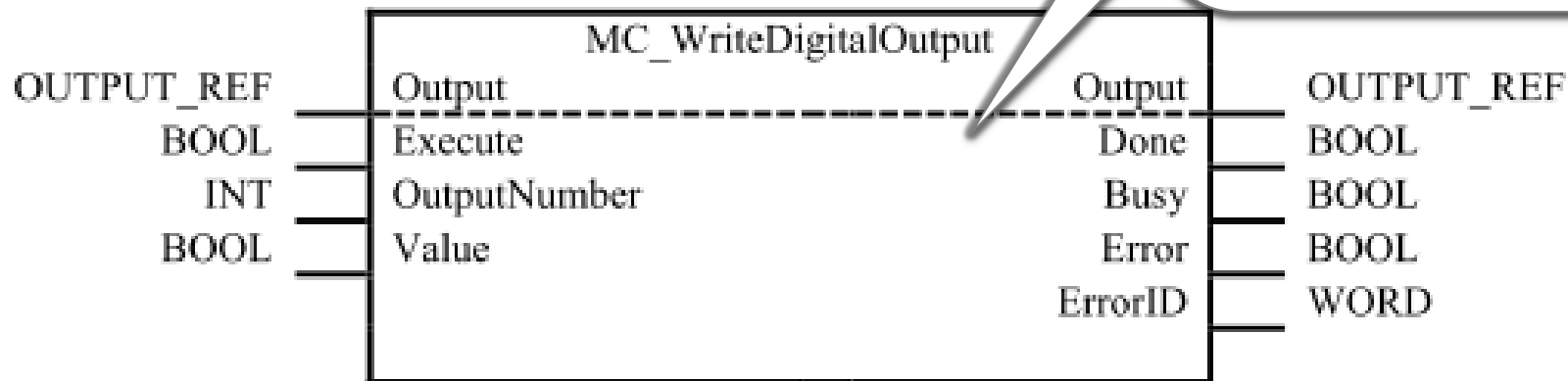
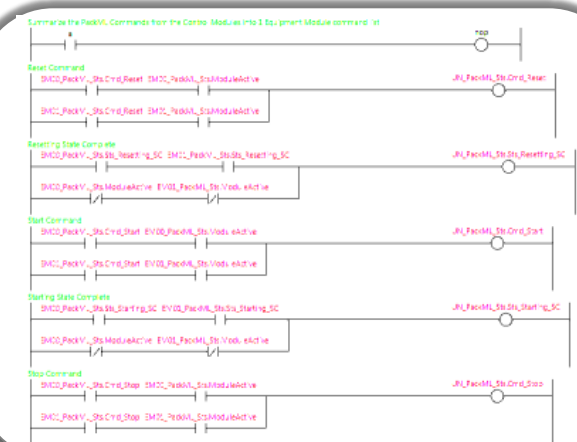
POU(プログラム構成単位) とは、

- プログラム
- ファンクションブロック(FB)
- ファンクション (FUN)

の総称。(※第3版ではPOUの1つに“クラス”も追加)

名称と 入力／出力となる変数が 明確に定義され、
内部のロジックは外部から隠されている。

内部ロジック



モーション制御ファンクションブロックの例

旧来：巻物ラダー回路のブツ切り

IEC：POUを用いた階層構造化

と言っているものの、実際に動作する機器制御を
巧く構造化・モジュール化実装できるのか？
フレキシブルなスマート生産設備への対応に役に立つのか？
→ 検証を実施しSCF2017展に展示しました

機能の区切りはどこ？
入力・出力はどれ？
全体が良く判らない！

各機能の入力/出力が明確
機能の階層構造が明確

● 構造化・モジュール化によるエンジニアリング効率化

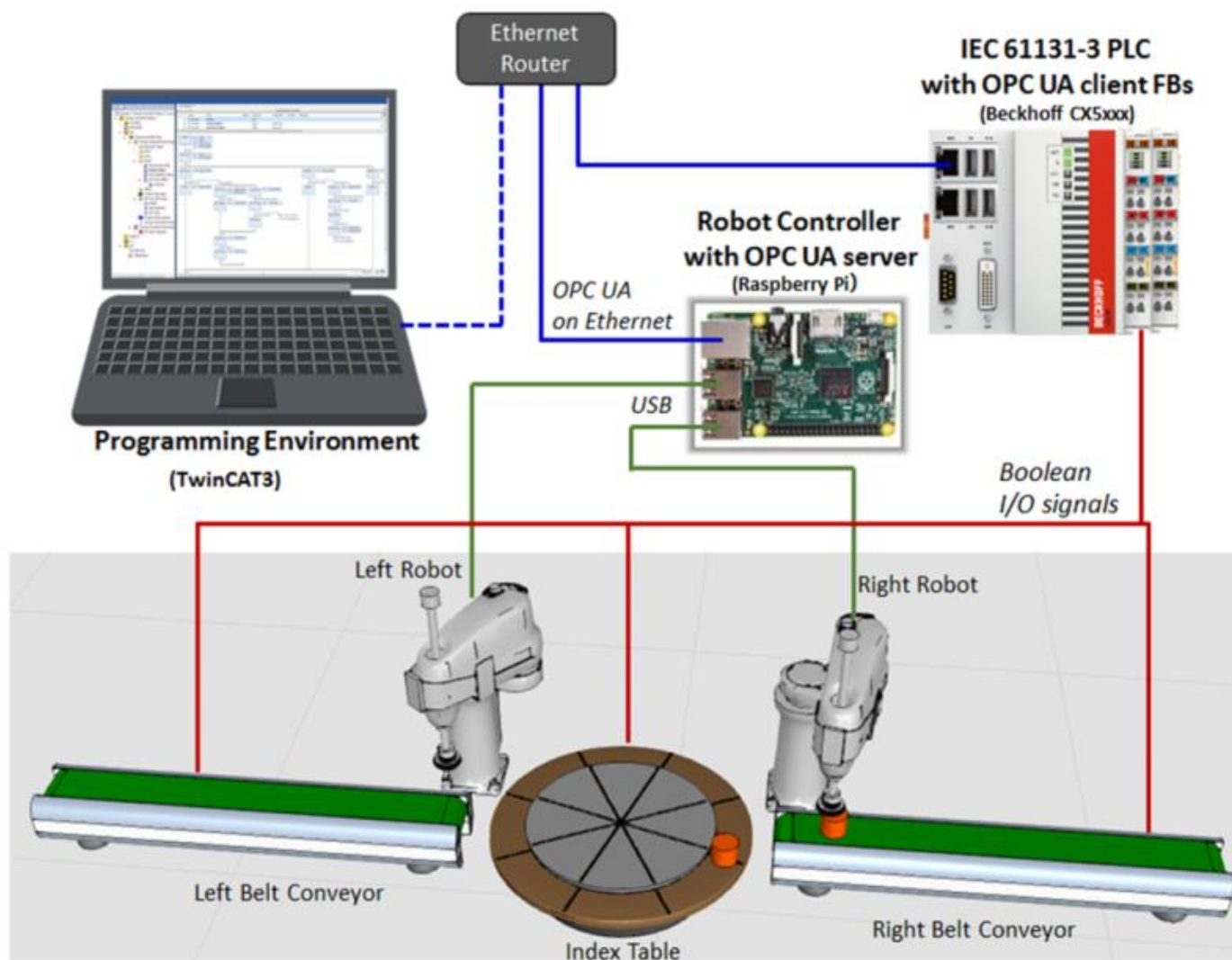
- 理解しやすい小さい単位に機能分割し、疎結合にしてモジュール化
- モジュール単位で単体テストを行い上流で品質確保
- モジュール単位で複数人並行開発を実施
- 動作記述に適切にSFC（シーケンシャル・ファンクション・チャート）を活用
- IEC 61131-3 第3版(2013)新言語機能の活用検討（※IAF CLiC観点では割愛）

● スマート生産設備（多品種生産、ロボット活用）挙動への対応

- 検証デモ設備にロボットを活用し、多品種生産やマスカスタマイズに対応できるようなPLC制御プログラムのモジュール化構造を検討する。

● 制御層の業界標準仕様を連携された広義なモジュール化検討（※IAF CLiC観点で後づけ）

- PLC制御プログラムのみならず、メカ・エレキ構成までを含めたモジュール化の検討
- KPI計算のために必要だがPLC制御プログラムしか知りえない装置の内部状態量について、上手に取得するための仕掛けをモジュールに用意できるか。



多品種対応設備で活用されることが多いロボットをシステムに取り入れる。

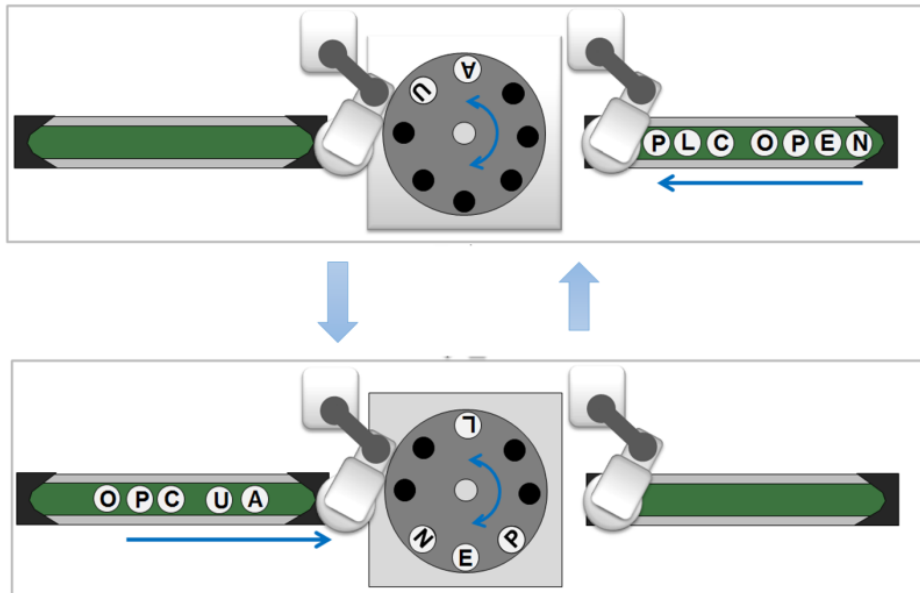
ロボットは、ロボットコントローラ(専用ソフトウェアが動作するRaspberry Pi)で制御されるSCARAタイプ。

ロボットコントローラにOPC UAサーバを搭載し、OPC UAクライアントから動作コマンドを受けられるように構成。

検証装置の動作仕様 – マスカスタマイズ対応

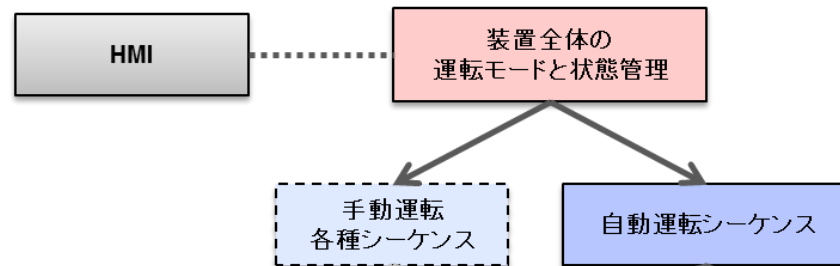
マスカスタマイズ的な動作仕様：

センサ入力に対して条件反射的に既定の動作を繰り返すような大量生産向けの制御方式ではなく、上位系から指示した語順となるように他方のコンベア上にワークを並び変える動作仕様を作成。



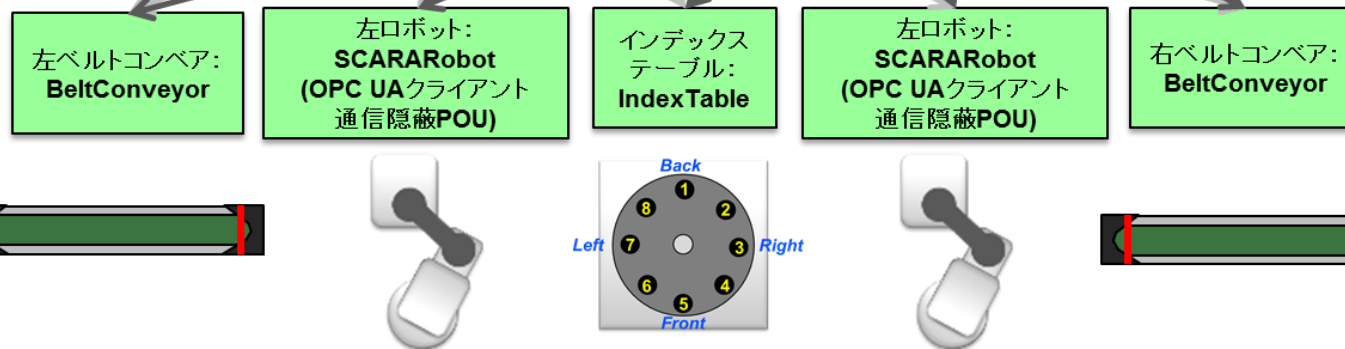
1. 装置に共通の運転状態管理と上位I/F

ラインを構成する装置に共通で適用されるべき 運転状態の管理、標準HMI、Safety制御などを実装したモジュール。
エンドーザ固有の方針や、PackML、MT Connectといった業界標準に従って決定されることが多い。



2. 装置や製造品種に固有のロジック

下位の機構モジュールが提供する機能を呼び出して、ある品種の加工や製造、特殊なメンテナンス動作などを実現するロジックを実装したプログラム・モジュール



3. 各機構モジュール制御

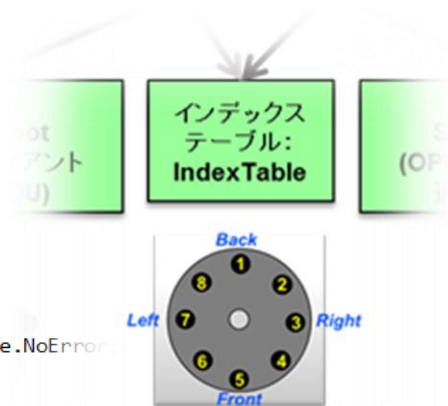
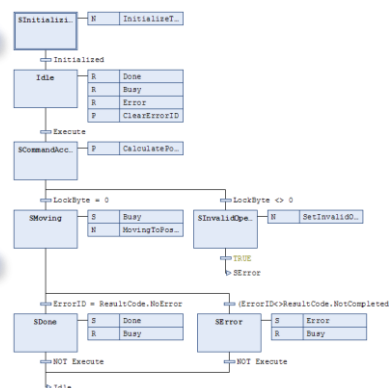
特定のメカ機構とセンサ・アクチュエータを制御して、特定の機能を実現するための機構制御ロジックが実装されたプログラム・モジュール。

インデックステーブル制御FB ベルトコンベア制御FB の実装例

インデックステーブル制御FBの機能

- ・ 位置指定回転（インデックス番号、目標位置）
- ・ テーブル位置リセット・ワーク情報リセット
- ・ ワーク情報記録（スロット番号、ワークオブジェクト参照）
- ・ ワーク探索（文字列、スロット番号）
- ・ ワーク情報削除（スロット番号）
- ・ ジョグ（方向）

```
FUNCTION_BLOCK PUBLIC IndexTable
VAR_INPUT
    Execute : BOOL := FALSE;
    TargetSlot : SlotID := 1;
    TargetPos : Position := 1;
END_VAR
VAR_OUTPUT
    Done : BOOL := FALSE;
    Busy : BOOL := FALSE;
    Error : BOOL := FALSE;
    ErrorID : ResultCode := ResultCode.NoError;
END_VAR
END_FUNCTION_BLOCK
```

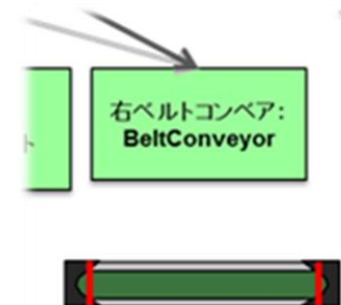


**メカ機構特有の詳細なI/O制御を隠蔽し、
各提供機能の呼出し手段を提供**

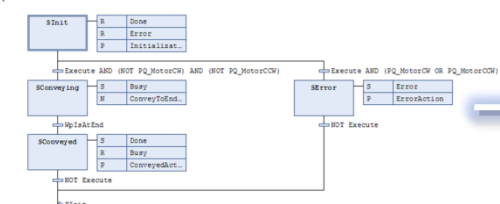
ベルトコンベア制御FB機能

- ・ 末端にワークが検出されるまで搬送
- ・ 指定時間搬送
- ・ ジョグ

```
(*
* 指定方向へ指定時間動作する。
*)
FUNCTION_BLOCK BeltConveyor_MoveTime
VAR_INPUT
    Execute : BOOL;
    Direction : ConveyDir;
    Duration : TIME;
END_VAR
VAR_OUTPUT
    Done : BOOL;
    Busy : BOOL;
    Error : BOOL;
    ErrorID : ResultCode;
END_VAR
(*
* 指定方向へ動作し、ワーク検出センサがワークを検出したら停止する。
*)
```



```
FUNCTION_BLOCK BeltConveyor_ConveyWpToEnd
VAR_INPUT
    Execute : BOOL;
    ConveyDir : ConveyDir;
END_VAR
ErrorID : ResultCode;
END_VAR
```



SCARAロボットFBの機能

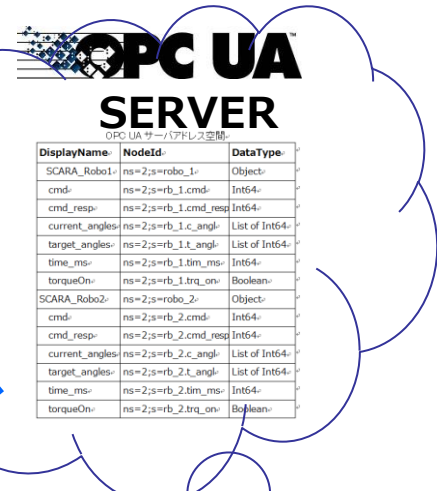
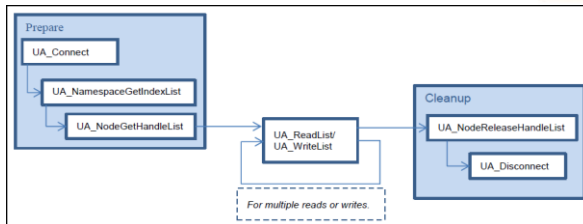
- 原点復帰
- インデックステーブルからのピックアップ
- コンベアへのプレイス
- コンベアからのピックアップ
- インデックステーブルへのプレイス



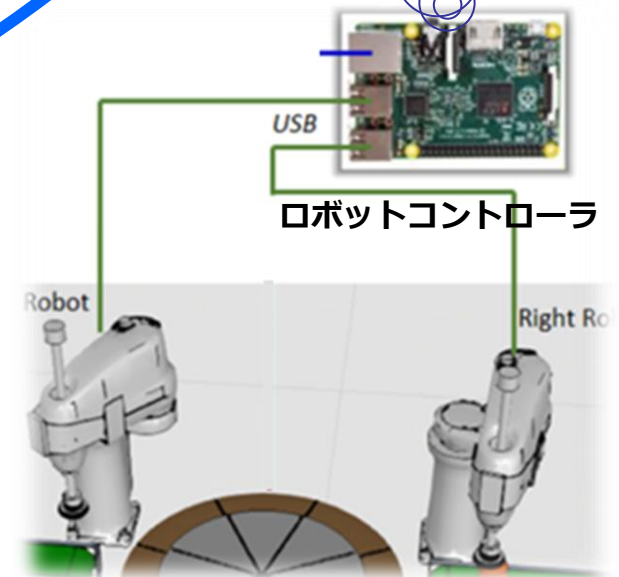
**OPC UAクライアント通信手続きを隠蔽して
ロボットの各動作呼出し機能を提供**

OPC UAクライアント通信FB

OPC FoundationとPLCopenが
共同で策定した通信FB群



DisplayName	NodeId	Data Type
SCARA_Robo1	ns=2;s=robo_1	Object
cmd	ns=2;s=rb_1.cmd	Int64
cmd_resp	ns=2;s=rb_1.cmd_resp	Int64
current_angles	ns=2;s=rb_1.c_angle	List of Int64
target_angles	ns=2;s=rb_1.t_angle	List of Int64
time_ms	ns=2;s=rb_1.tim_ms	Int64
torqueOn	ns=2;s=rb_1.trq_on	Boolean
SCARA_Robo2	ns=2;s=robo_2	Object
cmd	ns=2;s=rb_2.cmd	Int64
cmd_resp	ns=2;s=rb_2.cmd_resp	Int64
current_angles	ns=2;s=rb_2.c_angle	List of Int64
target_angles	ns=2;s=rb_2.t_angle	List of Int64
time_ms	ns=2;s=rb_2.tim_ms	Int64
torqueOn	ns=2;s=rb_2.trq_on	Boolean



1. 装置に共通の運転状態管理と上位I/F

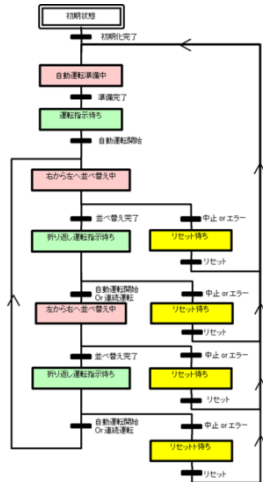
ラインを構成する装置に共通で適用されるべき 運転状態の管理、標準HMI、Safety制御などを実装したモジュール。
エンドーザ固有の方針や、PackML、MT Connectといった業界標準に従って決定されることが多い。



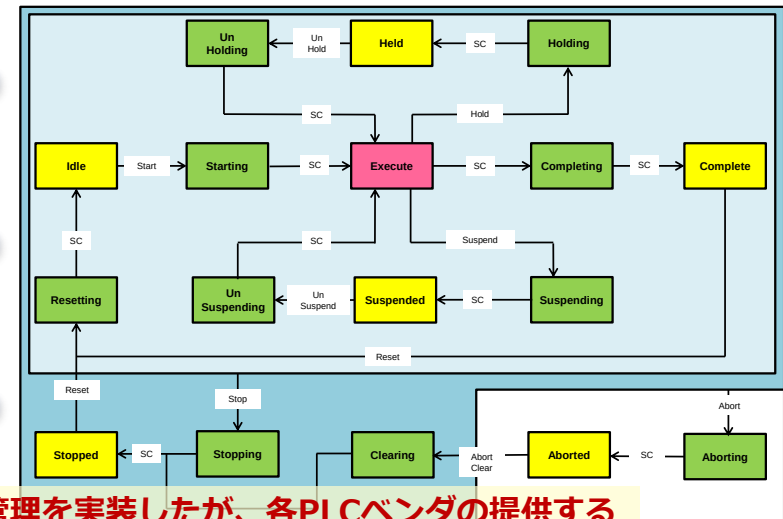
2. 装置や製造品種に固有のロジック

装置全体の運転状態管理FB

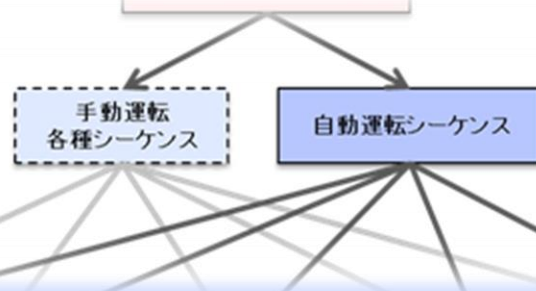
デモの自動運転の開始・停止、
手動操作モードなどを管理



PackML モード・ステート管理FB



今回はデモに最低限必要な状態管理を実装したが、各PLCベンダの提供するPackMLなど業界標準に対応したFBに置き換えることでOPC UAコンパニオンスペックへの対応が容易に。



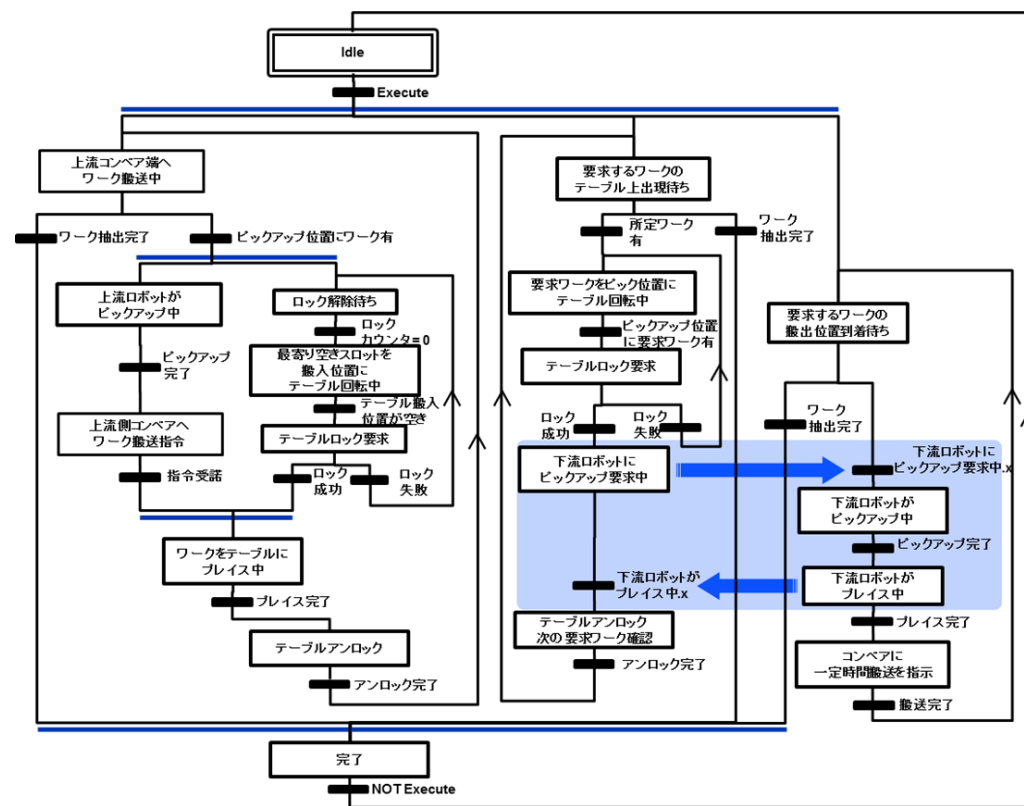
2. 装置や製造品種に固有のロジック

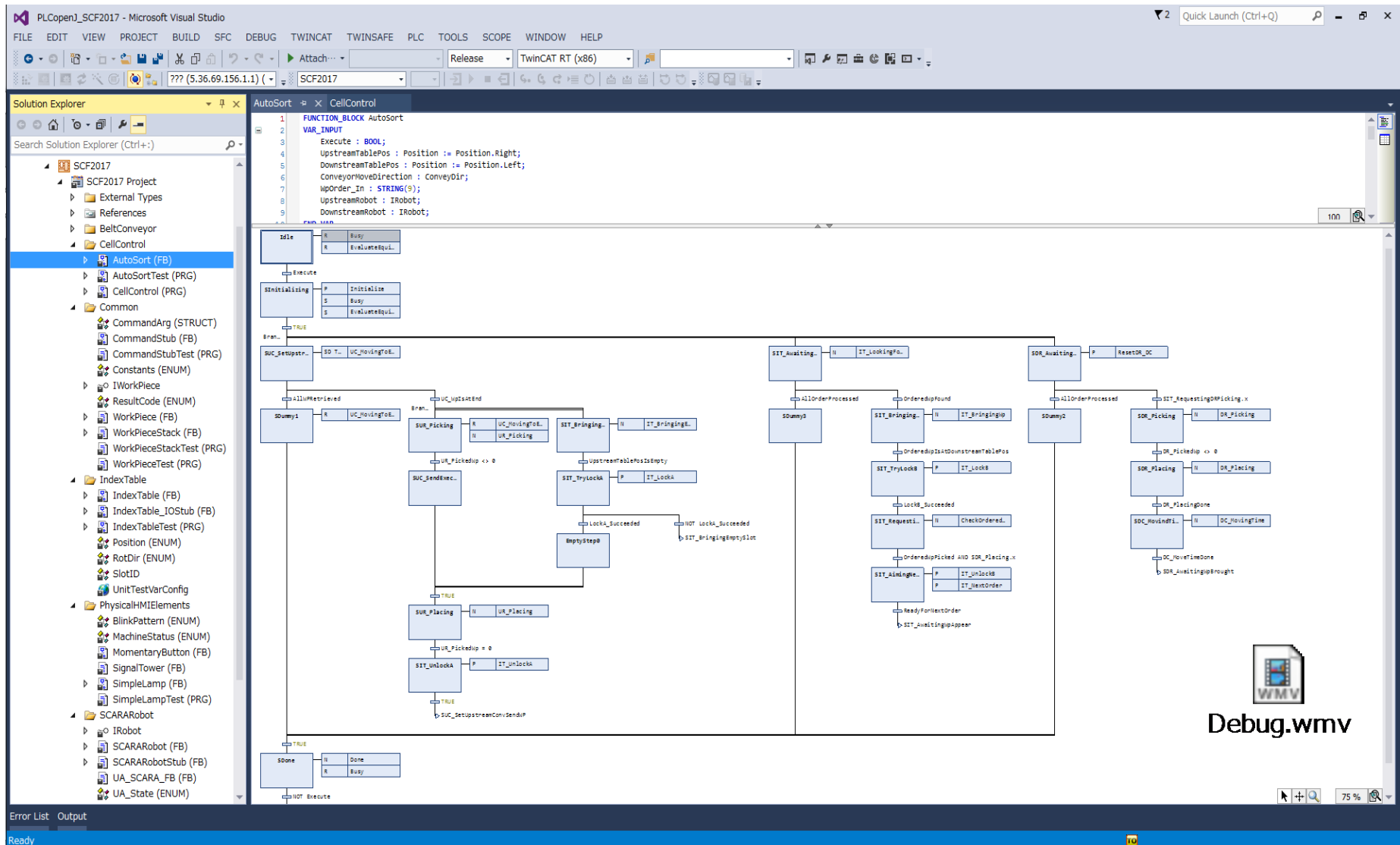
下位の機構モジュールが提供する機能呼び出して、
ある品種の加工や製造、特殊なメンテナンス動作など
を実現するロジックを実装したプログラム・モジュール

自動運転シーケンスFB

各機構モジュール（インデックステーブル、左右のロボット、左右のコンベア）の提供機能を呼び出して、指示された順にワークを並び替えるロジックを実装。

- 動作ロジックをSFCで設計し(右記)、そのまま動作するプログラムとして記述しデバッグできる
- このFBを別の入れ替えれることで、機構制御モジュールや運転状態管理モジュールはそのままに、全く異なる運転ロジックに対応することができる

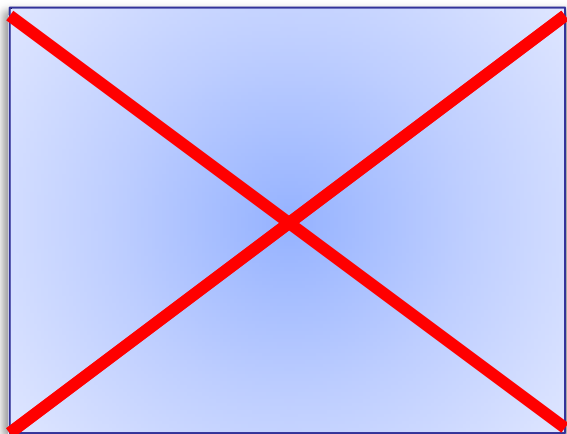




- **ロボットシステムとPLC制御機器で構成される検証装置を、実際に構造化・モジュール化して協調動作させることができた。**
 - ✓ ロボットコントローラに対し、PLCプログラムからOPC UAで動作指令を与えて協調動作を実現できた。
 - ✓ OPC UAクライアント・ファンクションブロックを用いた複雑な通信手続きは、PLC制御プログラムのロボット制御ファンクションブロック内に隠蔽できた。
 - ✓ 制御ソフトウェアを、運転状態管理層、アプリケーションロジック層、機構制御層の3層に関心を分離したモジュール構成で実現できた。
- **構造化・モジュール化とSFC(シーケンシャル・ファンクション・チャート)活用によるPLC制御ソフトウェア開発の効率化が実感できた。**
 - ✓ モジュール単位で遠隔地の4名で分担して効率的に並行開発できた。
 - ✓ SFCを用いて動作仕様が見える形で実装することで容易にデバッグできた。
 - ✓ モジュール単位で単体テストを行うことで、実設備なしのプログラミング段階で品質を確保できた。結果、約1日で実設備での動作確認を完了した。（実設備で見つかったロジック不具合2件のみ）
- **SFCについては、活用上の注意点や課題も見つかった**
 - ✓ 検証レポートについては、PLCopen Japan Websiteで公開予定。

IAF CLiC(制御層における標準仕様連携) の活動 **PLCopen** for efficiency in automation

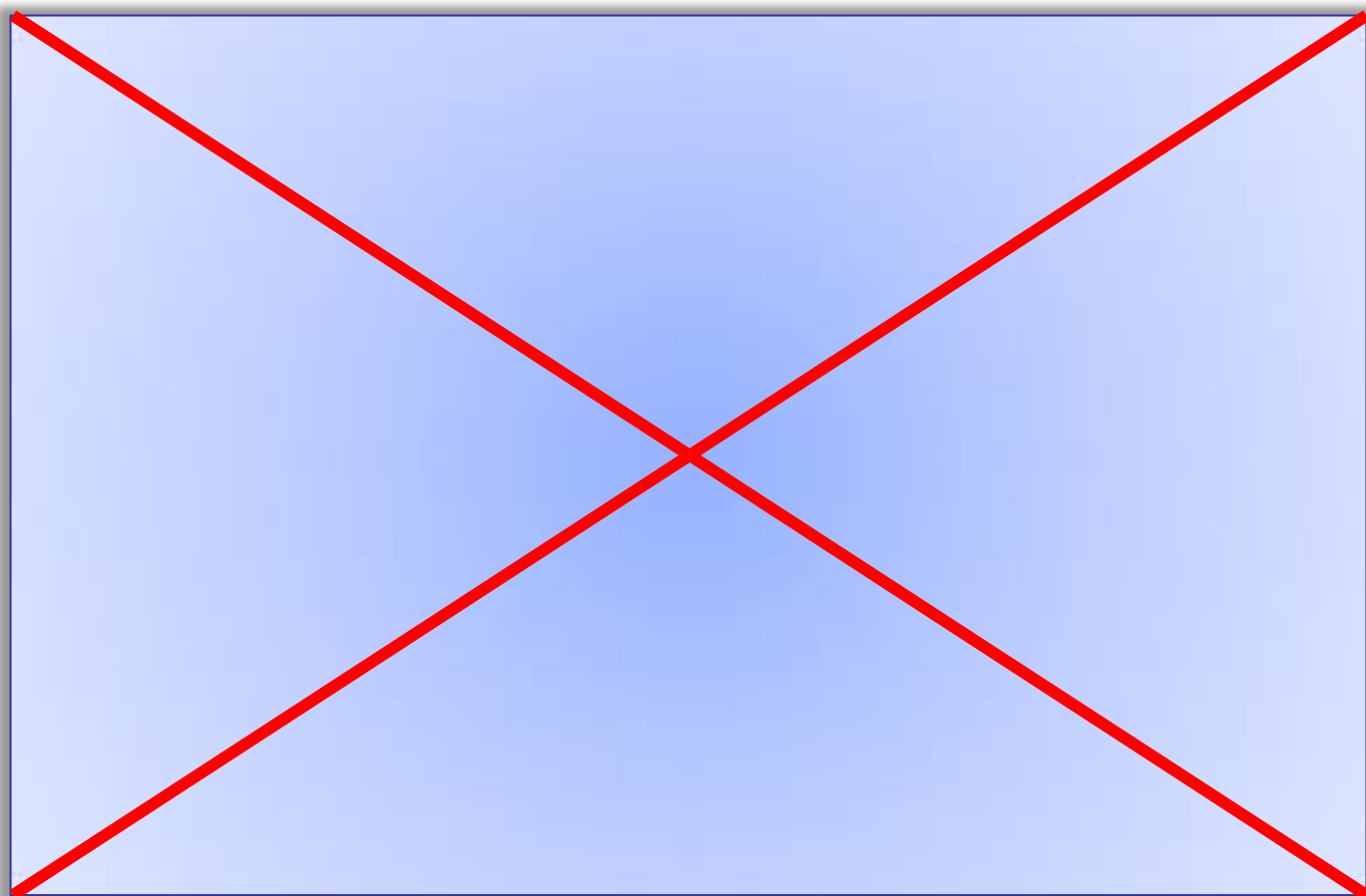
- OPC UA, AutomationML, PLCopen XML, FDT の仕様連携による、エンジニアリングドメインを超えた連携の可能性を継続検討 – モジュール化、KPIへの活用など



※OPC Foundation Websiteより転載

AutomationMLは、設計・制作フェーズにおけるエンジニアリングデータ連携手段を、

OPC UAは、運用フェーズにおけるオンラインデータ連携手段を提供する。



※AutomationML Websiteより転載

http://www.plcopen.org/pages/whats_new/industry4.0.htm

PLCopen starts new working group

on PLCopen Functionalities in Industry 4.0

Industry 4.0, or Industrie 4.0, is a movement that has great impact on the world of industrial automation.

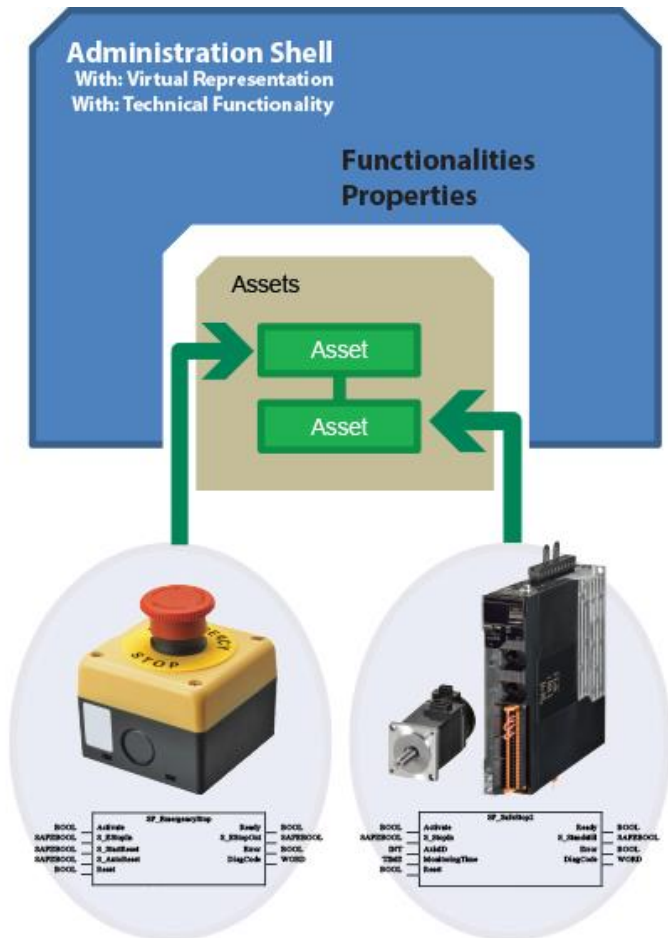
Within this activity, an Administration Shell has been defined (see picture below). Within this shell the physical aspects of the assets are defined in eCI@ass. However the functionalities of these assets and components are not defined, while many of these are specified by PLCopen, like done for motion control, safety and communication. These PLCopen specifications provide a well defined suite of functionalities, covering the basic range. In addition there is a framework defined by PLCopen related to the definition of function blocks with a focus to the definition of user specific libraries. With this one can encapsulate larger functionalities which are linked to the higher levels of integration. This helps to do the engineering of a new machine more easily and to provide a more decomposed and reusable structure in the software development, making it easier to add new wishes.

For this reason PLCopen starts a new working group to make a proposal to the Industry 4.0 initiative on how to map the existing PLCopen functionalities to the Industry 4.0 Administration shell. The kick-off meeting is set for December 12, 2017 in Modena, Italy. Contact PLCopen if you want to attend this.

PLCopen欧州本部でも Industry 4.0の文脈での

PLCopen仕様の活用を論じる作業部会が2017年末に発足。

Administration Shell の Asset におけるPLCopenロジック機能の活用・連携方法などについて議論中。



ご清聴ありがとうございました。



<http://www.plcopen-japan.jp/>

PLCopen[®] is a registered trademark owned by the association PLCopen, as well as the PLCopen logos