

初心者が

# オブジェクト指向(IEC61131-3 ed.3)で Pick and Place装置作ってみた

第15回FA設備技術勉強会  
2023/09/03

yuyu (@yuyuTds)

# 自己紹介

- ▶ 名前 : yuyu (@yuyuTds)



Twitter (X)



本日の資料



- ▶ 仕事 : IEC61131-3 ed.2を使ってラダー、ST  
を書いてます
- ▶ FA歴 : 1.5年
- ▶ 初めてのLTです。  
"FA\_Study"とタグを付けてツイートしてください！

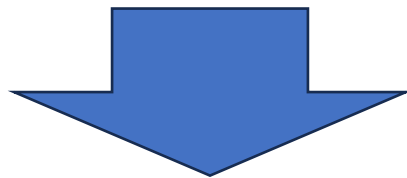
# きっかけ



オブジェクト指向は聞いたことがあるけど、  
実際にどのように装置を作るのかイメージがつかない。



FA設備技術勉強会で発表してみたい！



オブジェクト指向(IEC61131-3 ed.3)を使って、  
Pick and Place装置作ってみよう！！

# 本編に入る前に... オブジェクト指向とは？

オブジェクト指向は、データとそれに関連する操作を1つの「オブジェクト」と呼ばれる単位にまとめ、複雑なソフトウェアをモジュール化し、保守性や再利用性を向上させるための手法です。

- **オブジェクト（インスタンス）**：プログラム内の部品です。例えば、モーターやセンサーのようなものです。それぞれが特定の役割と情報を持ちます。
- **クラス（FB）**：オブジェクトの設計図です。同じ種類のオブジェクトが持つ共通の属性と操作を定義します。例えば、モーターのクラスには「速度」や「開始」などの属性や操作が含まれます。
- **カプセル化**：オブジェクトは、外部から見える部分（操作）と見えない部分（データ）を持ちます。これにより、安全にオブジェクトを操作できます。
- **継承**：新しいオブジェクトを作るとき、既存のクラスを基にして作成できます。新しいオブジェクトは、元のクラスの特性を受け継ぐことができます。
- **ポリモーフィズム**：同じ操作が異なるオブジェクトで異なる方法で機能することを意味します。これにより、複数のオブジェクトを柔軟に扱えます。

# 作った装置



家庭用WindowsPC



Windows上で  
動くPLC



OPC-UA

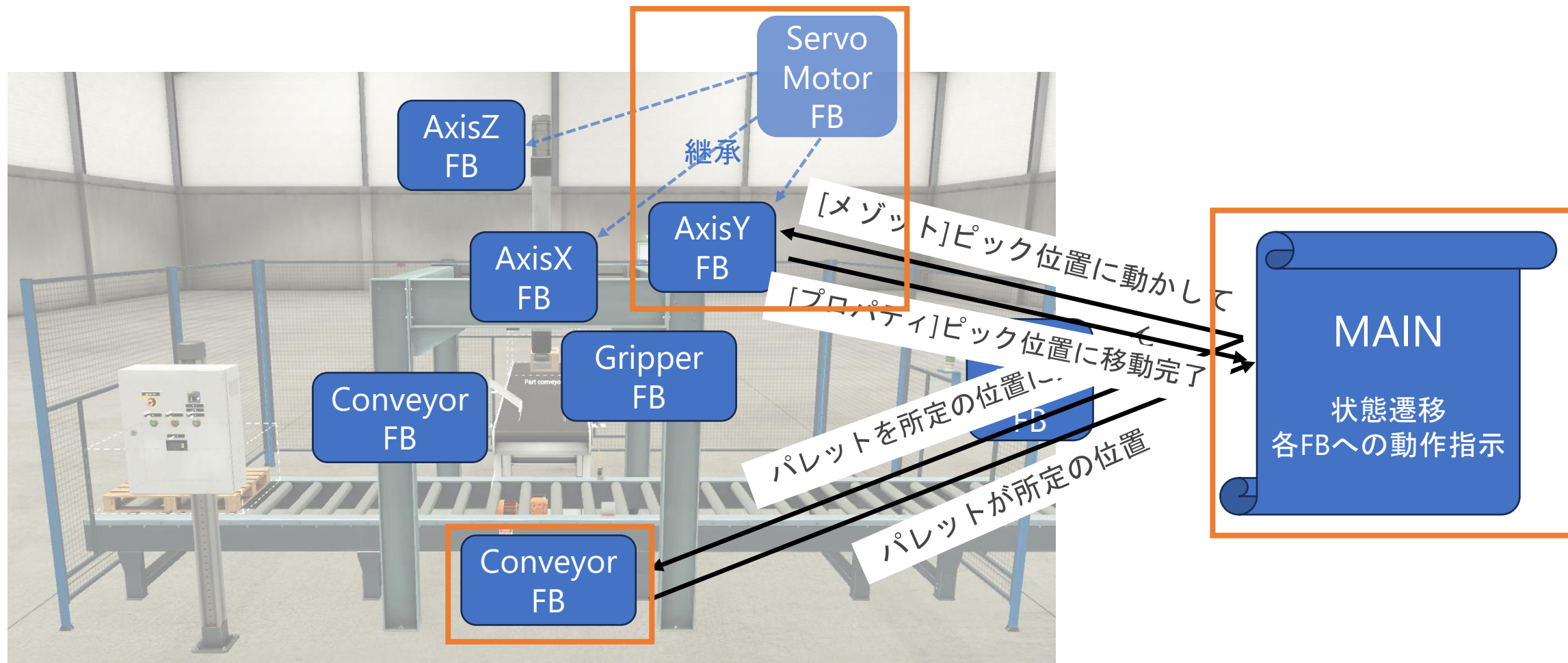


シミュレー  
ションソフト

FACTORY I/O

# プログラム構成

アクチュエータごとに7つのFBとMAINで構成



# FBの中身 [fbConveyor]

## プロパティ（状態取得）

- AtPlace
- AtNotPlace

ワークが所定の位置にあるときにTRUE

ワークが所定の位置にないときにTRUE

## メソッド（動作指示）

- Set
- Stop
- Discharge

ワークを所定の位置まで移動する

コンベアを即停止する

ワークを排出する



# FBの中身 [fbAxisX]

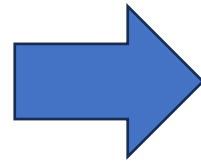
## fbServoMotor

プロパティ（状態取得）

メソッド（動作指示）

- CallFBs
- Disable
- Enable
- Home
- MoveAbsolute
- MoveRelative
- Stop
- Reset

継承



## fbAxisX

プロパティ（状態取得）

- IsInitPos
- IsPickPos
- IsPlacePos1
- IsPlacePos2
- IsPlacePos3

メソッド（動作指示）

- MoveInit
- MovePick
- MovePlace1
- MovePlace2
- MovePlace2



# FBの中身 [fbAxisY、fbAxisZ]

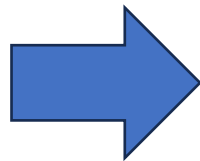
## fbServoMotor

プロパティ（状態取得）

メソッド（動作指示）

- CallFBs
- Disable
- Enable
- Home
- MoveAbsolute
- MoveRelative
- Stop
- Reset

継承



## fbAxisY

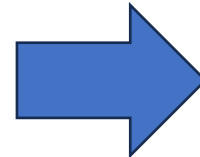
プロパティ（状態取得）

- IsInitPos
- IsPickPos
- IsPlacePos1
- IsPlacePos2
- IsPlacePos3

メソッド（動作指示）

- MoveInit
- MovePick
- MovePlace1
- MovePlace2
- MovePlace2

継承



## fbAxisZ

プロパティ（状態取得）

- IsInitPos
- IsPickPos
- IsPlacePos1
- IsPlacePos2
- IsPlacePos3

メソッド（動作指示）

- MoveInit
- MovePick
- MovePlace1
- MovePlace2
- MovePlace2

# MAINの中身

```
1 AxisX();
2 AxisY();
3 AxisZ();
4 PalletConveyor(Sensor=>GVL.SimCmd.PalletPlace, Inverter=>GVL.SimCmd.PalletConveyor);
5 WorkConveyor(Sensor=>GVL.SimCmd.PartPlace, Inverter=>GVL.SimCmd.PartConveyor);
6 Gripper(Cplus=>GVL.SimCmd.CPlus, Adsorb=>GVL.SimCmd.Grab);
7 StackLight(RedLight=>GVL.SimCmd.ExitRed, YellowLight=>GVL.SimCmd.ExitYellow, GreenLight=>GVL.SimCmd.ExitGreen);
8
9 IF NOT (GVL.SimSts.Stop) THEN
10     StepNo := 900;
11 END_IF
12
13 IF StepNo = 0 THEN
14     StackLight.Stopped();
15
16     IF GVL.SimSts.Reset THEN
17         StepNo := 100;
18     END_IF
19
20 ELSIF StepNo = 100 THEN
21     StackLight.Resetting();
22     Gripper.UnGrip();
23     Gripper.UnRotate();
24
25     IF TRUE THEN
26         StepNo := 110;
27     END_IF
28
29 ELSIF StepNo = 110 THEN
30     AxisX.Enable();
31     AxisY.Enable();
32     AxisZ.Enable();
33
34     IF NOT (AxisX.Busy) AND NOT (AxisY.Busy) AND NOT (AxisZ.Busy) THEN
35         StepNo := 120;
36     END_IF
37
38 ELSIF StepNo = 120 THEN
39     AxisX.Home();
40     AxisY.Home();
41     AxisZ.Home();
```

FB自体をサイクリックに呼び出す

ステップ番号を使って  
状態遷移  
&  
各FBに動作指示

# MAINの中身 [ピック部分抜粋]



```
131 ELSIF StepNo = 290 THEN
132     AxisX.MovePick();
133     AxisY.MovePick();
134
135     IF AxisX.IsPickPos AND AxisY.IsPickPos THEN
136         StepNo := 300;
137     END_IF
138
139 ELSIF StepNo = 300 THEN
140
141     IF WorkConveyer.AtPlace THEN
142         StepNo := 310;
143     END_IF
144
145 ELSIF StepNo = 310 THEN
146     AxisZ.MovePick();
147
148     IF AxisZ.IsPickPos THEN
149         StepNo := 320;
150     END_IF
151
152 ELSIF StepNo = 320 THEN
153     Gripper.Grip();
154
155     IF TRUE THEN
156         StepNo := 330;
157     END_IF
158
159 ELSIF StepNo = 330 THEN
160     AxisZ.MoveInit();
161
162     IF AxisZ.IsInitPos THEN
163         StepNo := 340;
164     END_IF
```

- } 動作指示：X軸とY軸をピック位置に移動
- } ステップ遷移条件：X軸とY軸がピック位置
- } ステップ遷移条件：ワークがセット完了
- } 動作指示：Z軸をピック位置に移動
- } ステップ遷移条件：Z軸がピック位置
- } 動作指示：ワークを吸着
- } ステップ遷移条件：なし
- } 動作指示：Z軸を初期位置に移動
- } ステップ遷移条件：Z軸が初期位置

# 感想

- PLCはモノを制御するので、オブジェクト指向と相性がよい！
  - FBによる階層化で、MAINプログラムはかなり分かりやすく書くことができる。
  - FBの継承は、非常に強力！
  - FBによって、シーケンス処理と並列処理が書きやすい。
- 
- インストールから完成まで約20時間。
  - 初めてのツールということもあり難しかった。  
まだまだオブジェクト指向を使いこなせていない。。

# 感想

- TwinCATの情報はネット上に溢れており、検索すると結構目的のサイトが見つかる！
- Twitter(X)上でいろいろな人が助けてくれました。ありがとうございました。



# 最後に

- [こちら](#)でソースコード、立ち上げ方法を公開しています。無料でできるので、是非試してみてください！  
また、コメントも頂けると非常に嬉しいです。

ありがとうございました！

# 参考にさせていただいたサイト

- [FactoryIO#Beckhoff TwinCAT3と連携しよう | SOUP01](#)
- [Beckhoff#TwinCAT3 TF6100 OPCUA Part1 Server立ち上げよう | SOUP01](#)
- [Beckhoff#DUT詳しく見よう | SOUP01](#)
- [Beckhoff#TwinCAT3 Function Block-1 | SOUP01](#)
- [Beckhoff#TwinCAT3 Function Block-2 | SOUP01](#)
- [TwinCAT Howto](#)
- [2.1. 便利なデバッグ補助機能 — TwinCATテクニカルノート 第1版 ドキュメント](#)
- [Beckhoff Information System – English](#)
- [SPT Application Framework Documentation](#)