

Function Blocks for Motion Controlの 標準化

–PLCopen Japan Motion Control WGの取り組み–



PLCopen Japan
Motion Control WG 相川 富士雄
2007. 11. 16
(株)安川電機 モーションコントロール事業部

1. 仕様制定のコンセプト
2. 技術仕様の種類と状況
3. Function Blocksの実装
4. 開発環境
5. 動作例

プログラミング言語の標準化 <IEC61131-3を利用>



Function Blocks for Motion Control を提唱

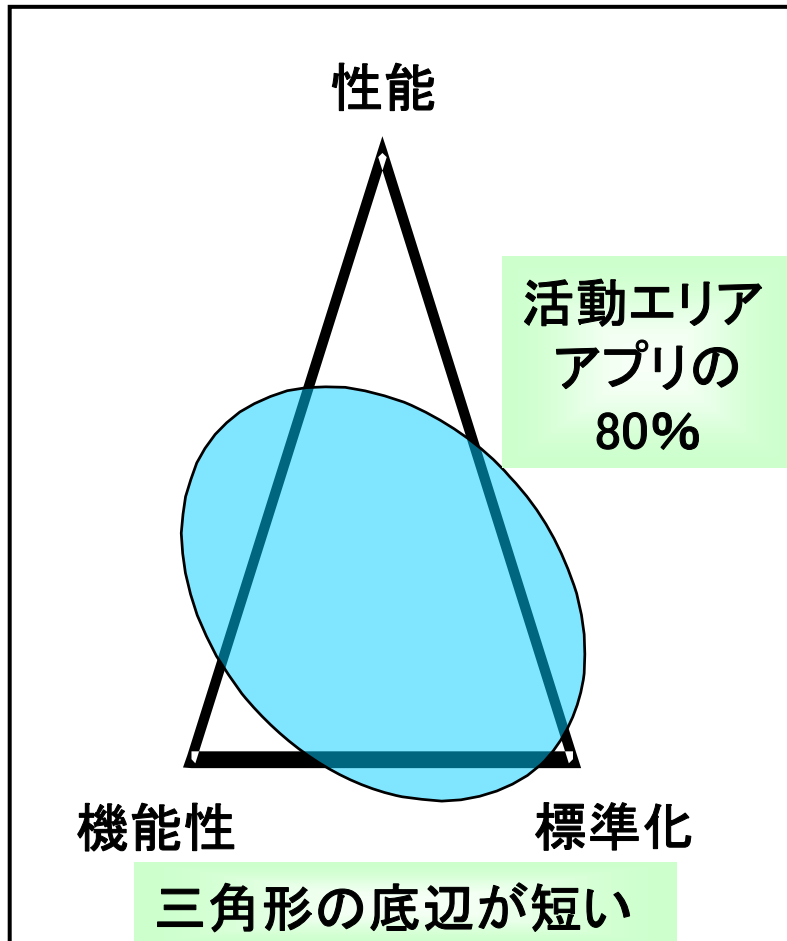
仕様だけでなく、プログラムのインターフェースまで定義(標準化)

ハードウェアへの依存性を低減

- ⇒ アプリケーションソフトウェアの再利用性を向上
- ⇒ トレーニングコストを低減
- ⇒ 様々なアプリケーションに対応
<5つのPartで仕様を構成し、単軸～多軸～補間機能を実現>

1. 仕様制定のコンセプト(2)

ユーザの3つの選択肢



- ・性能の追求：
ハードウェアに密接なプログラム
- ・機能の拡充：
ユーザにとっては非常に有用
- ・標準化：
トレーニングコストを最少化

本仕様の位置付け

- ⇒ アプリの80%カバー
- ⇒ 三角形の底辺が短い

2. 技術仕様の種類と状況(1)

5つのPartで規定している仕様の内容と各々の関係

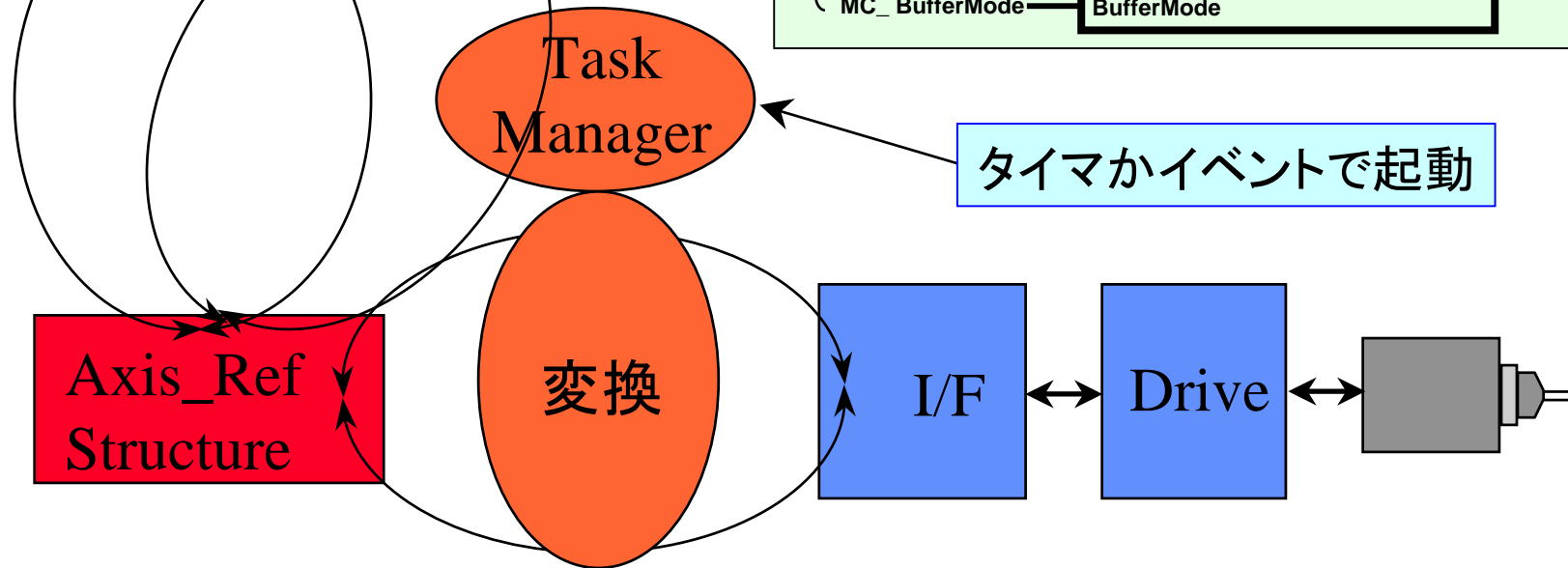
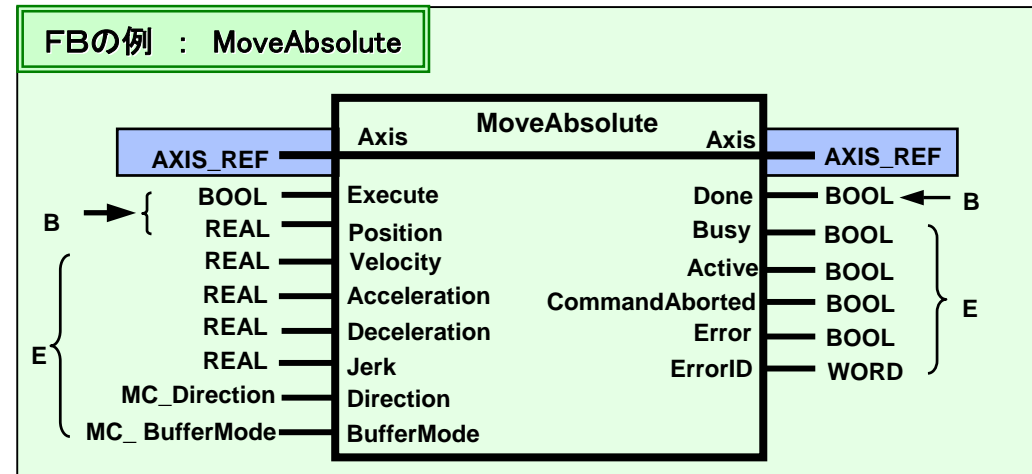
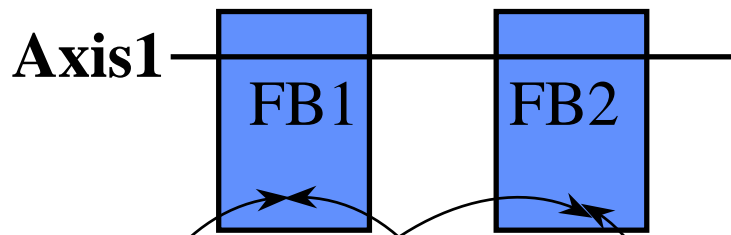
- ✧ Part 1 – Function Blocks for Motion Control
＜基本仕様＞
- ✧ Part 2 – Extensions
＜Part1からの拡張仕様＞
- ✧ Part 3 – User Guidelines
＜ユーザ使用から見たガイドライン＞
- ✧ Part 4 – Interpolation
＜多軸間の仕様＞
- ✧ Part 5 – Homing
＜原点サーチ関連の追加仕様＞

各Partのリリース状況

- ✧ Part 1 – Function Blocks for Motion Control
 - <Ver1. 1 : 2005/04/09 リリース済>
 - <Ver1. 0 : 2001/11/23 リリース済>
- ✧ Part 2 – Extensions
 - <Ver1. 0 : 2005/09/16 リリース済>
- ✧ Part 3 – User Guidelines
 - <Ver0. 3 : 2004/04/16 リリース>
- ✧ Part 4 – Interpolation ⇒ <Draft制作中>
- ✧ Part 5 – Homing
 - <Ver0. 99: 2005/11/10 リリース>
 - [2006/04/30:フィードバックコメント]

3. Function Blocksの実装

アプリケーションからハードウェアへの依存性を低減する仕組み



シーケンスと同じ開発環境で、モーションのアプリケーションを開発

[システム構成]

汎用PLCシステムを利用してモーション制御システムを構築

従来 : モーション制御用のアプリケーションプログラムを開発する
専用の開発環境が必要

↓

<IEC61131-3のプログラミング言語を採用>

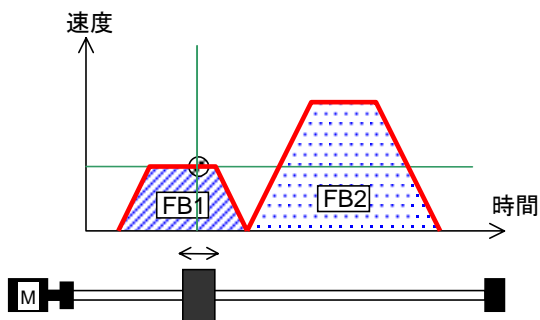
↓

シーケンス制御用のアプリケーションプログラムを開発するのと同じ
開発環境を使用可能

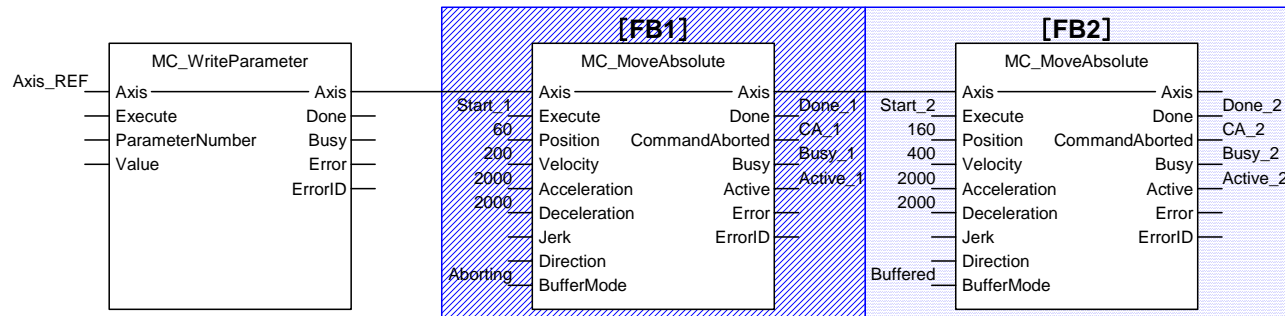
5. 動作例 <単軸動作: Demo#1>

Part1で規定 ⇒ 単独軸において、モード指定によりFB間の動作の違いを実現

Buffered Mode

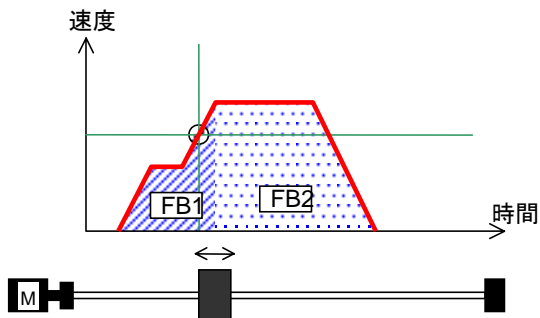


<Buffered Modeの Program例>

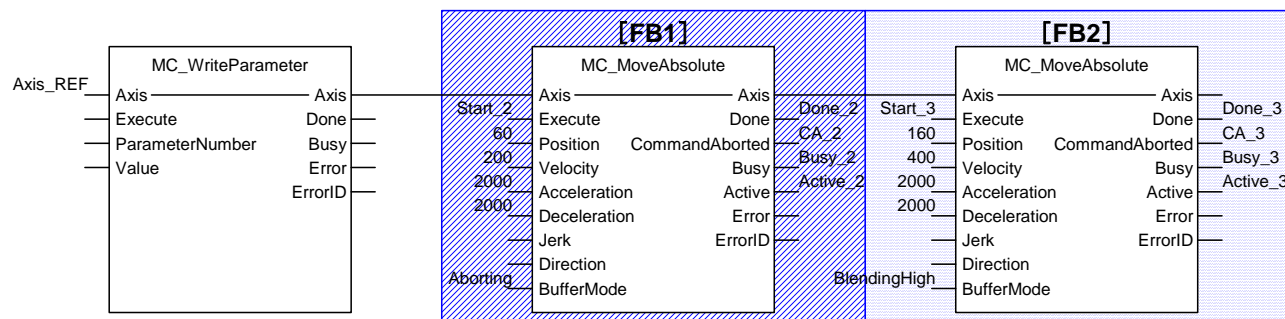


※[FB1]の動作が完了すると、直ぐに[FB2]が動作

Blending High Mode



<Blending High Modeの Program例>

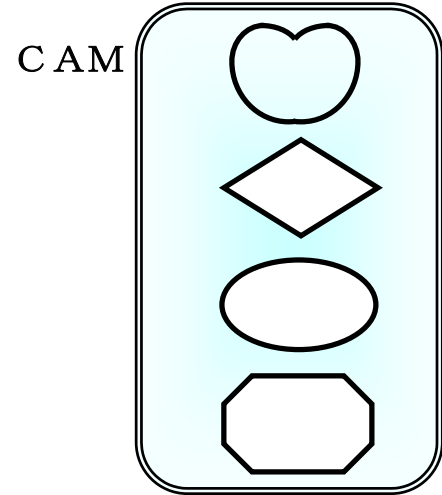
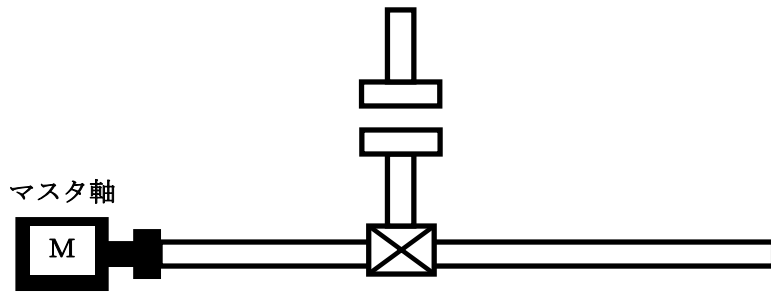


※[FB1]の最終位置では、[FB2]の速度(速い方)を使用して連続的に動作

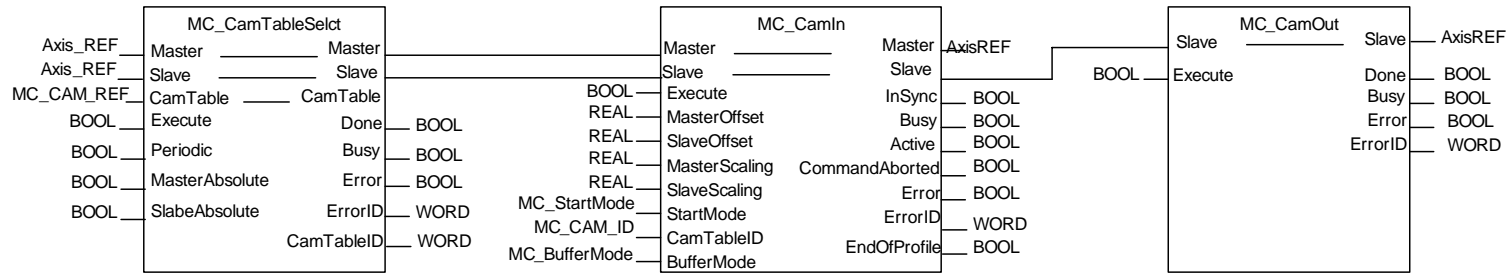
5. 動作例 <CAM動作: Demo#2-1>

Part1で規定 ⇒ 機械式CAMによる運転パターンを、電気式のカム動作で実現

1) 停止時



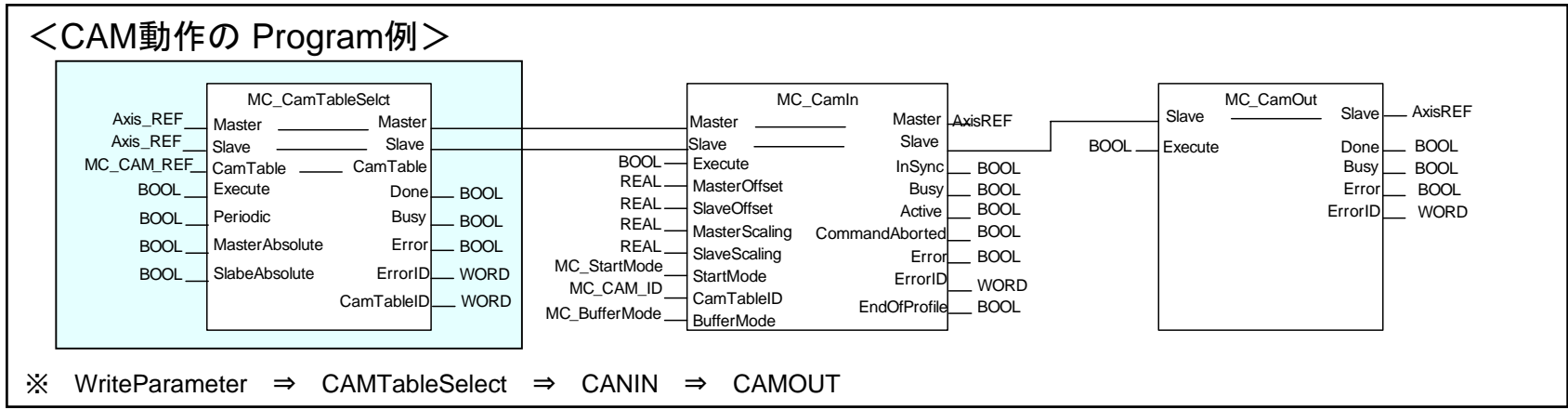
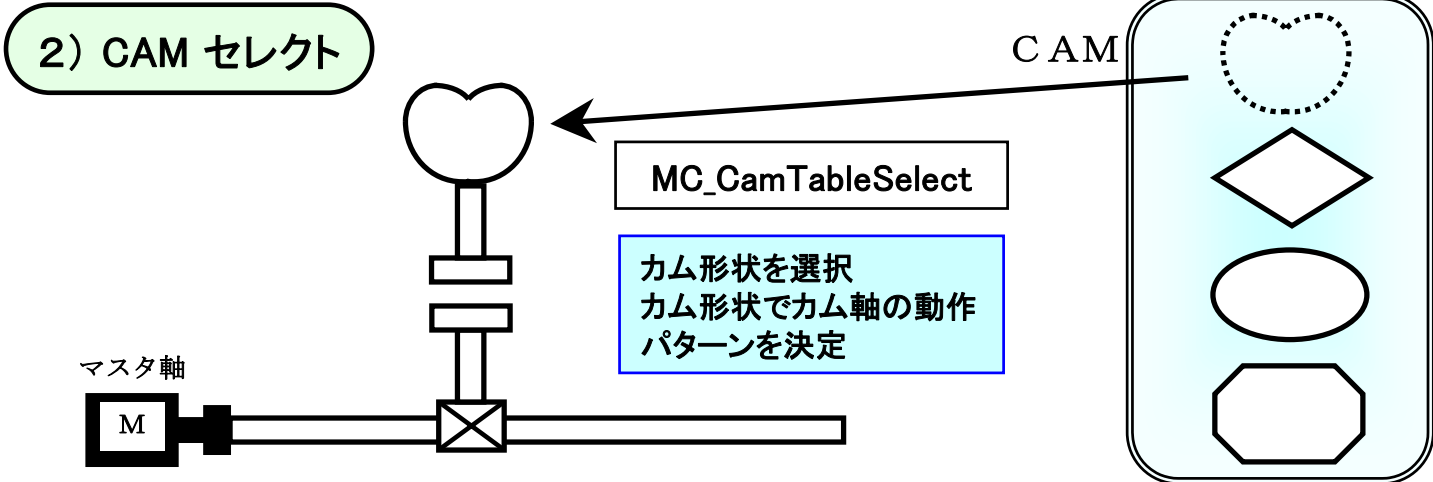
<CAM動作の Program例>



※ WriteParameter ⇒ CAMTableSelect ⇒ CANIN ⇒ CAMOUT

5. 動作例 <CAM動作: Demo#2-2>

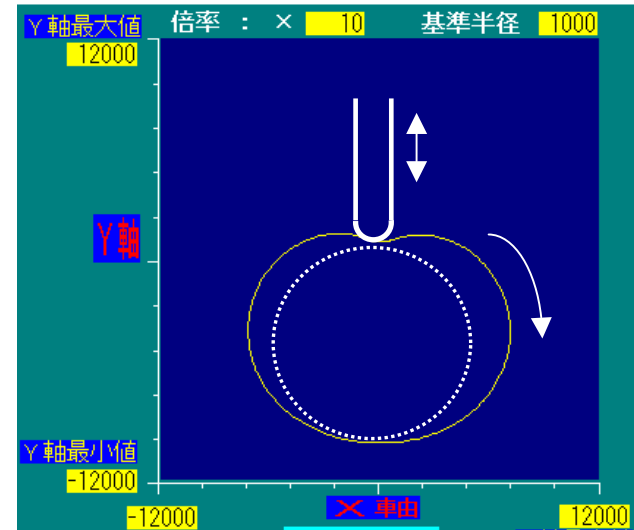
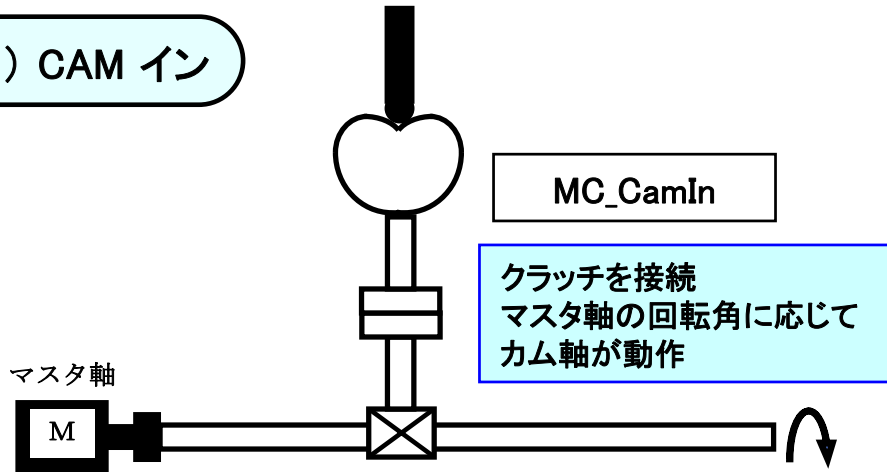
Part1で規定 ⇒ 機械式CAMによる運転パターンを、電気式のカム動作で実現



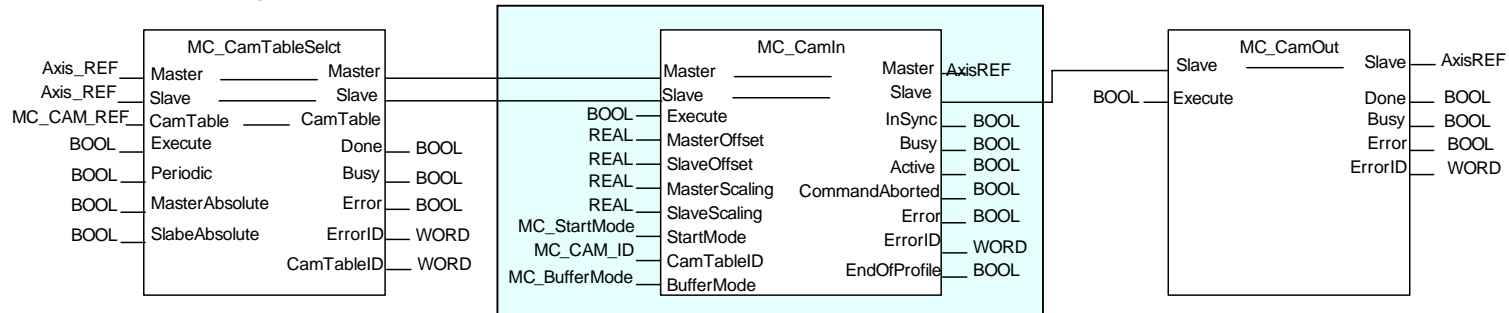
5. 動作例 <CAM動作: Demo#2-3>

Part1で規定 ⇒ 機械式CAMによる運転パターンを、電気式のカム動作で実現

3) CAM イン



<CAM動作の Program例>

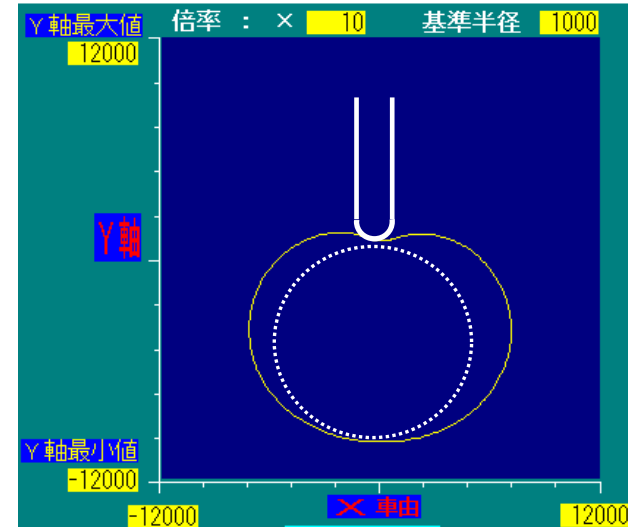
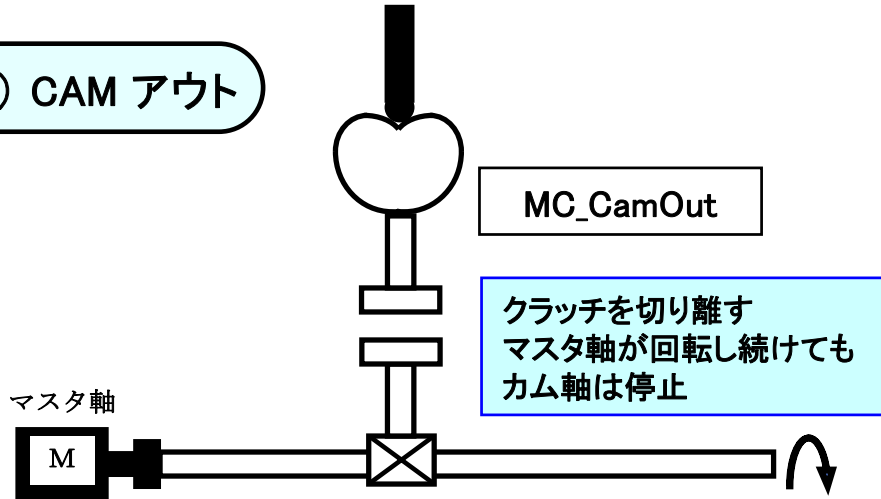


※ WriteParameter ⇒ CAMTableSelect ⇒ CANIN ⇒ CAMOUT

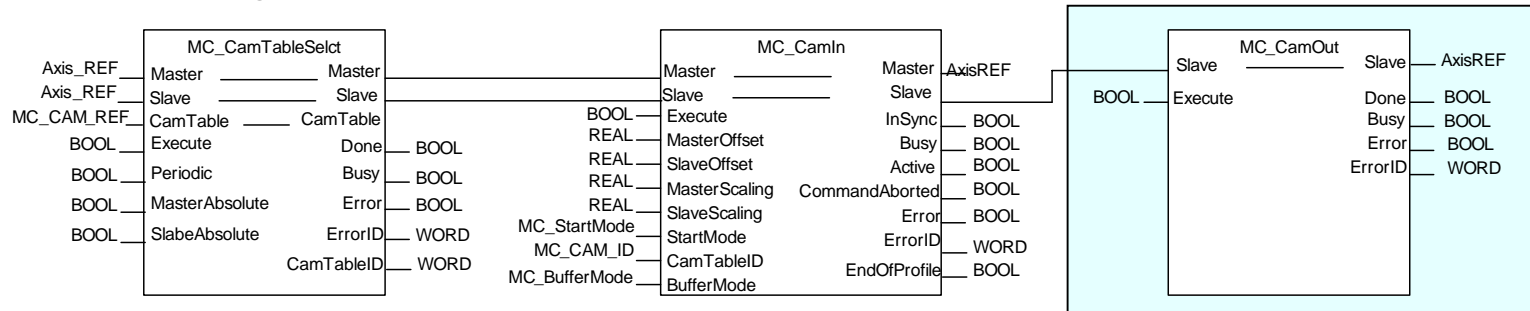
5. 動作例 <CAM動作: Demo#2-4>

Part1で規定 ⇒ 機械式CAMによる運転パターンを、電気式のカム動作で実現

4) CAM アウト



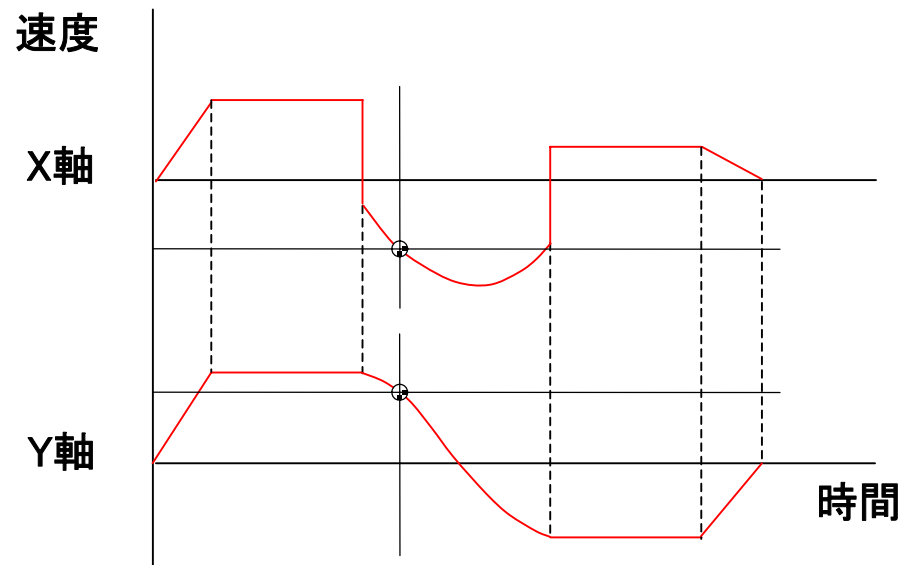
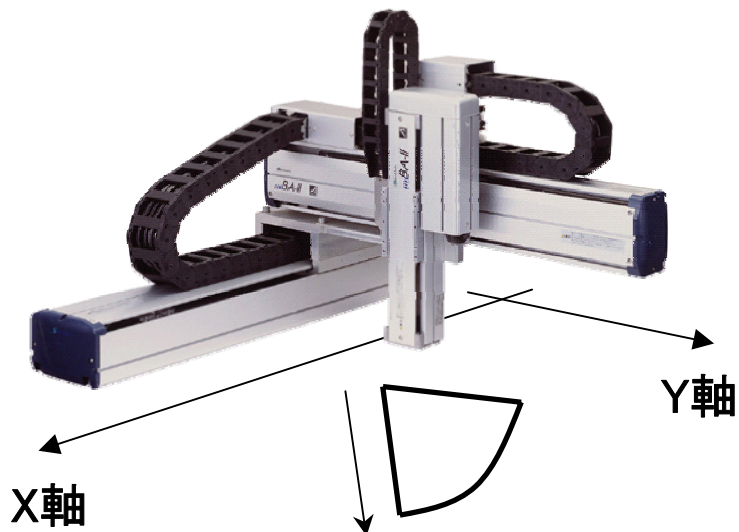
<CAM動作の Program例>



※ WriteParameter ⇒ CAMTableSelect ⇒ CANIN ⇒ CAMOUT

5. 動作例 <補間動作: Demo#3>

Part4で規定 ⇒ マスタースレーブの関係でなく、複数軸で関連しあった補間動作を実現



<補間動作の Program例>

AxGroupXY	MC_MoveLinearAbsolute	AxisGroup	AxisGroup	
Go	Execute	Done	Done1	
[300;100]	Positions	Busy	Busy1	
1000	Velocity	Error	Error1	
100	Acceleration	ErrorID	ErrID1	
100	Deceleration			
Blending	TransitionMode			

	MC_MoveCircularAbsolute	AxisGroup	AxisGroup	
Done1	Execute	Done	Done2	
CENTER	CircMode	Busy	Busy2	
[300;100]	AuxPoint	Error	Error2	
[200;350]	EndPoint	ErrorID	ErID2	
1000	Velocity			
100	Acceleration			
100	Deceleration			
Blending	TransitionMode			

	MC_MoveLinearAbsolute	AxisGroup	AxisGroup	
Done2	Execute	Done	Done3	
[200;350]	Positions	Busy	Busy3	
1000	Velocity	Error	Error3	
100	Acceleration	ErrorID	ErrID3	
100	Deceleration			
Blending	TransitionMode			

※ 直線動作 ⇒ 円弧動作 ⇒ 直線動作