

# PLCのOSSってなに？

Programmable  
Logic  
Controller

×

Open  
Source  
Software

第18回FA設備技術勉強会  
2024/04/07

yuyu (@yuyuTds)

# 自己紹介

- 名前 : yuyu (@yuyuTds)
- 仕事 : PLCメーカーのFAE (SE)
- FA歴 : 2年 (24歳)
- ひとこと : 2回目の発表になります

#FA\_Studyでポストしてください！



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)

PLCのOSSをイメージできて

これいいじゃんって思ってもらおう

# OSS（オープンソースソフトウェア）とは？

- 誰でも自由に使用、変更、配布が可能なソフトウェア
- ソフトウェアのソースコードが公開
- 開発がコミュニティによって行われる

開発者は自分の開発したコードを公開。他の人もそれを使ったり、さらに改良を加えて、協力してソフトウェアを育てる

- 例：Firefox   LibreOffice   Linux

# OSSのメリット

- コスト：無料で利用可能
- 柔軟性：必要に応じてカスタマイズすることが可能
- 品質：多くの人によってコードがチェックされるため、問題が見つかりやすく、迅速に修正される（場合による）
- サポート：コミュニティによるサポートが利用可。作成したプログラムのアドバイスがもらえることも。（場合による）

# FAにおけるOSSの活用状況



FAでもOSSが多く使われている！　けどPLC本体は...

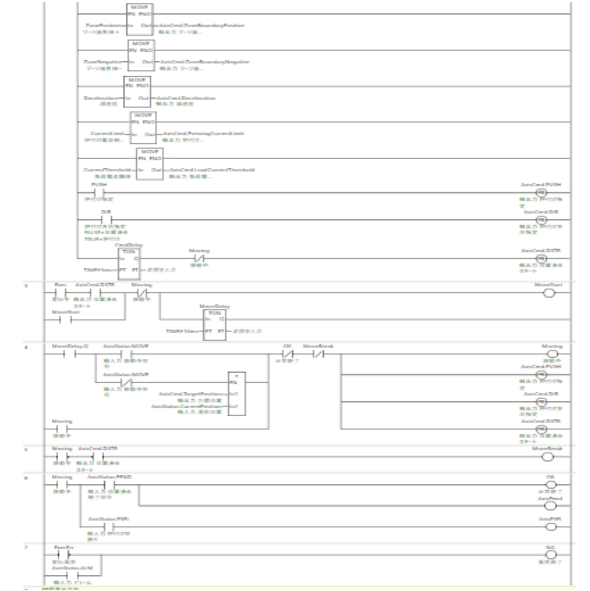
# 私の考えるPLCのOSSってどんなもの？

Q. IAIのロボシリンダを使って、2点間搬送をするプログラムをつくりたいと思ったらどうしますか？



# 私の考えるPLCのOSSってどんなもの？

- A. マニュアルを読んで、モータON、移動指示、目標位置、速度、加減速度などのアドレスを調べて、順序を守って通信するラダーを書く。



これって皆さん同じプログラムを作ってませんか？

プログラム共有したら開発効率が爆上がりするのでは！？

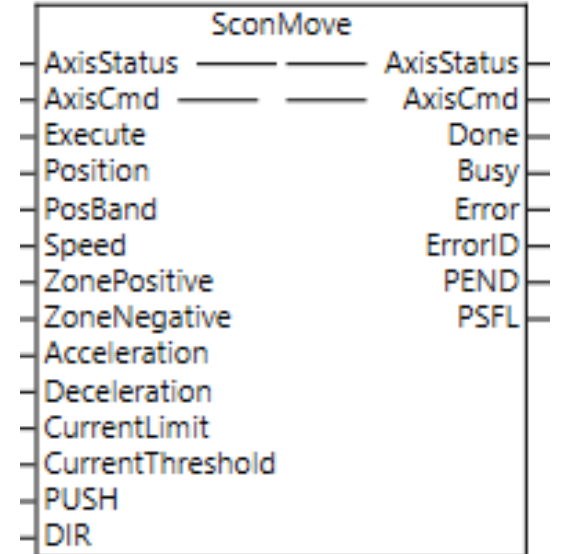
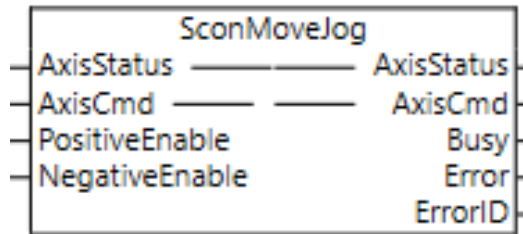
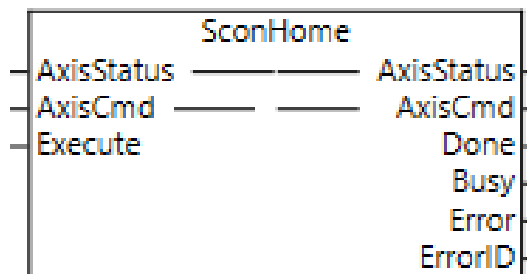
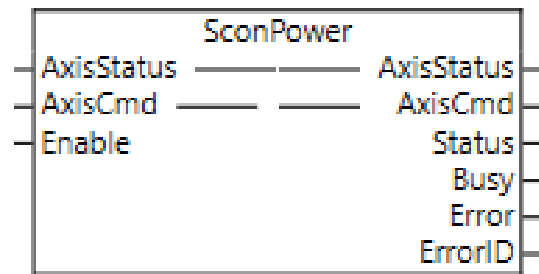


# 私の考えるPLCのOSSってどんなもの？

実はこのプログラムをOSSとして、公開してくれている方がいます！

OkitaSystemDesign

<https://www.osdes.com/oms/sample/other/SCON.html>



FBをコピーするだけで、SCONを使うことができる！

# 私の考えるPLCのOSSってどんなもの？

他にはどんなものが考えられる？

## 特定の機器との通信系

- ・ アクチュエータとの通信 (SCON)
- ・ センサとの通信 (画像機器)

## 便利ツール系

- ・ 数値計算 (微分、ATAN)
- ・ データ変換処理 (2の補数変換)

## アプリケーション系

- ・ データロギング (CSV保存)
- ・ エラー管理

## 上位アプリケーションとの通信系

- ・ DB系 (MQTT、InfluxDB)
- ・ 操作系 (WakeOnLAN)

# PLCのOSS 現状は？

- 三菱、オムロン、キーエンス：各PLCメーカーが便利プログラム、つなぎプログラムの配布をしているが、OSSではなく、パッケージ化されたもの（改造、改良不可）
- Codesys、TwinCAT：少しずつ始まってきている（GitHubで見かける）



SPT-Libraries



PackML\_PLC\_Example



PID\_controller



CODESYS\_Serial



CODESYS-MQTT



IcIs-twinCAT-motion

国内メーカーのPLCのOSSは全然発展していない...

# なぜOSSが流行らない？（難しいところ）

- 品質が保証されていないOSSは怖くて使えない
  - ➡ 考え方を変える、仕組みを入れる、コミュニティの力
- PLCメーカーごとに開発環境が異なり、互換性がない
  - ➡ IEC61131-10が救世主になる？
- プログラムがハードウェアと密接に結びついている
  - ➡ IEC61131-3によって、少しは改善する？

# とりあえずやってみました

指定した周期のクロック信号  
を出力するFBを[GitHub](#)で公開  
しています

よかったら使ってください！

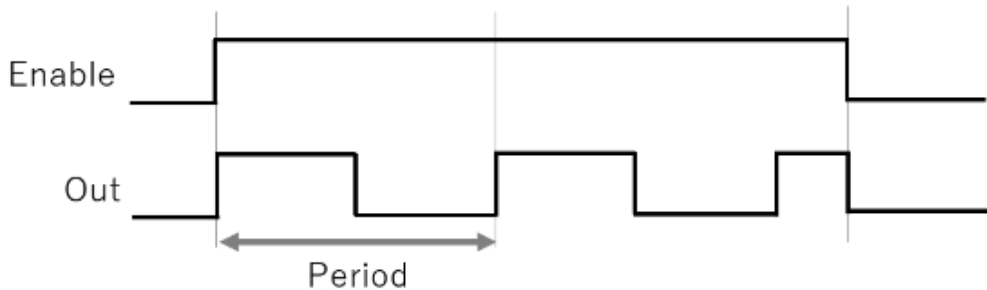


README MIT license

力							号です
---	--	--	--	--	--	--	-----

## タイミングチャート

- Enableの立上りで、OutはTRUEになります。（出力ONから始まります）
- Enable=FALSEで、OutはFALSEになります。



## 使用名前空間

みなさんもGitHubでOSSに参加しませんか？

# 最後に（お願い）

- ぜひ皆様のご意見を聞いてみたく、アンケートにご協力いただけると嬉しいです（選択式7問+自由記述(任意)）
- 結果は4/13(土)を目安に私のX（@yuyuTds）上で公開します！

アンケートフォーム



発表資料のリンク



ありがとうございました！ Thank you！

# [Appendix] ClockFBの使用方法

ClockFBを試みるには、オムロンのSysmacStudioが必要です。

体験版があり、30日間の無料で使用することができます。

以降、SysmacStudioの体験版をダウンロードする方法から説明します。

1. <https://license-user.automation.omron.com/>にアクセスして、ライセンスポータルに登録を行います。

参考動画：[https://www.fa.omron.co.jp/product/tool/ss-video-manual/ja/assets/movies/id\\_00129\\_JP.mp4](https://www.fa.omron.co.jp/product/tool/ss-video-manual/ja/assets/movies/id_00129_JP.mp4)

2. [ライセンスポータル](#)から、[Sysmac Studio 64bit] と [OMRON License Manager] をダウンロードして、ダウンロード後、インストールしてください。

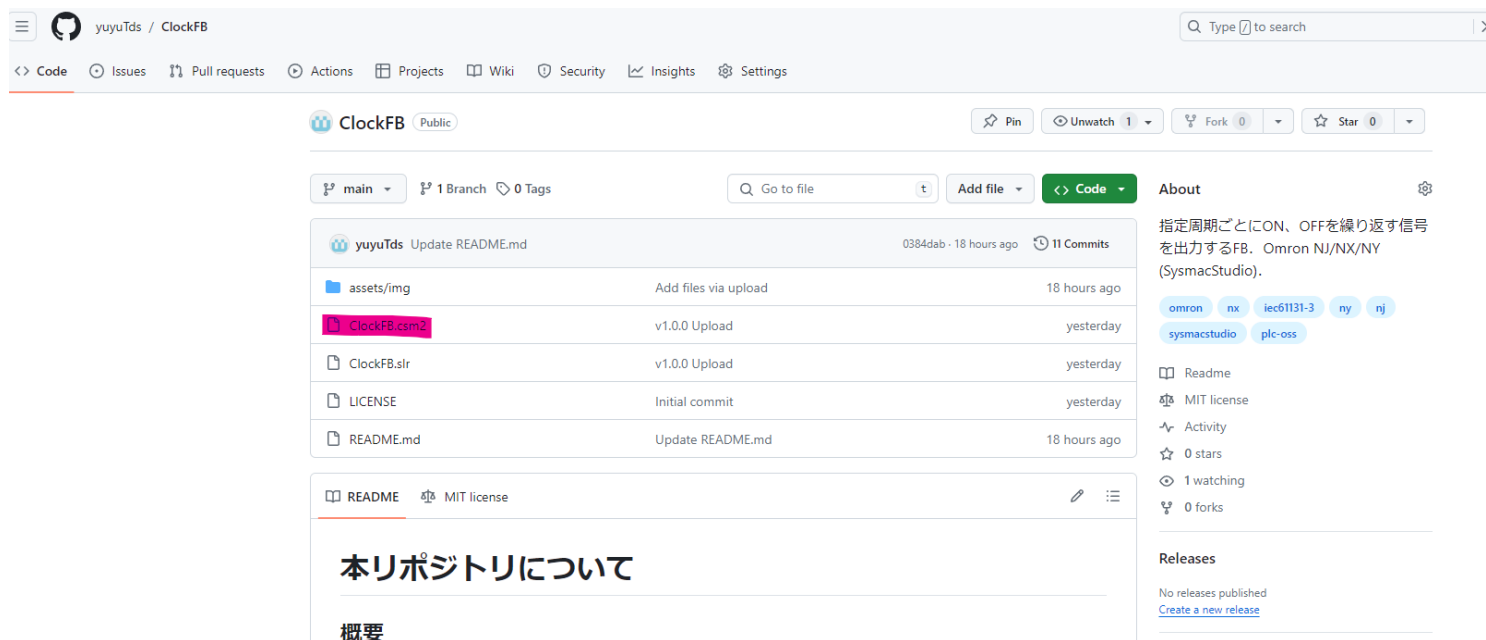
参考動画：[https://www.fa.omron.co.jp/product/tool/ss-video-manual/ja/assets/movies/id\\_00131\\_JP.mp4](https://www.fa.omron.co.jp/product/tool/ss-video-manual/ja/assets/movies/id_00131_JP.mp4)

SysmacStudio体験版のダウンロード、インストール作業は完了です。

# [Appendix] ClockFBの使用方法

次に、GitHubからClockFBをダウンロードする方法を説明します。

1. [GitHubのClockFBのリポジトリ](#)にアクセスします。
2. 今回ダウンロードしたい「ClockFB.csm2」をクリックします。



The screenshot shows the GitHub repository page for `yuyuTds / ClockFB`. The repository is public and contains the following files:

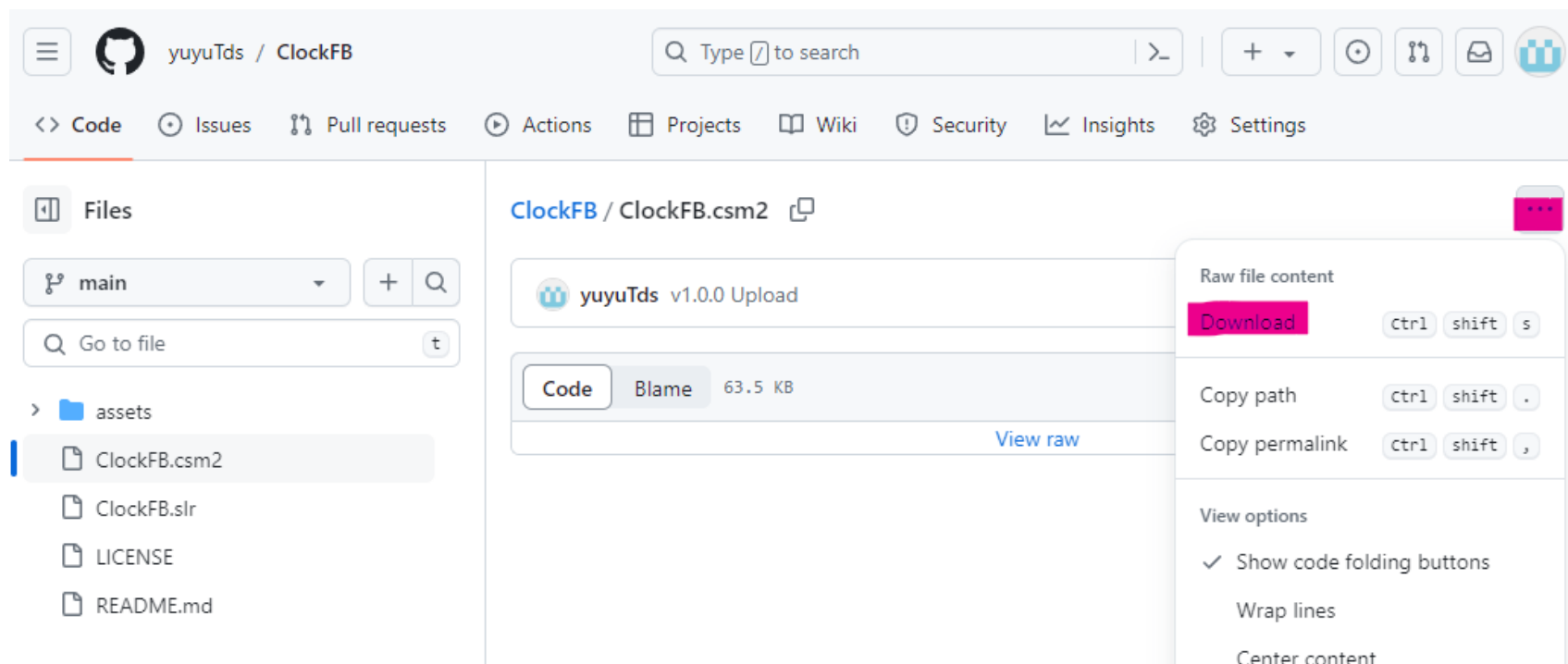
File	Commit	Time
assets/img	Add files via upload	18 hours ago
<b>ClockFB.csm2</b>	v1.0.0 Upload	yesterday
ClockFB.slr	v1.0.0 Upload	yesterday
LICENSE	Initial commit	yesterday
README.md	Update README.md	18 hours ago

The right sidebar shows the repository's description: "指定周期ごとにON、OFFを繰り返す信号を出力するFB。 Omron NJ/NX/NY (SysmacStudio)." and includes links to the README, MIT license, and Activity. The repository has 0 stars, 1 watching, and 0 forks.



# [Appendix] ClockFBの使用方法

3. 右上の「・・・」をクリックして、「Download」をクリックします。

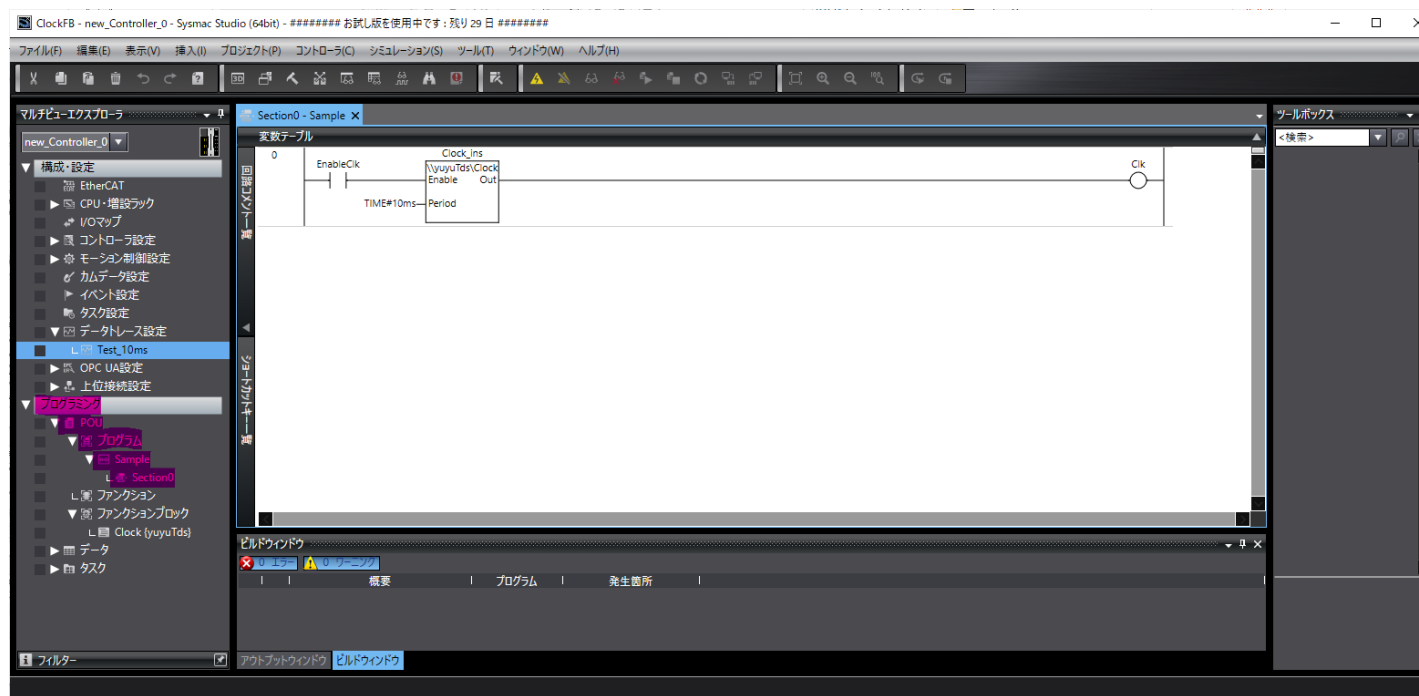


ClockFBのダウンロードが開始されます。

# [Appendix] ClockFBの使用方法

次に、ClockFBをシミュレーションで動かしてみる方法を説明します。

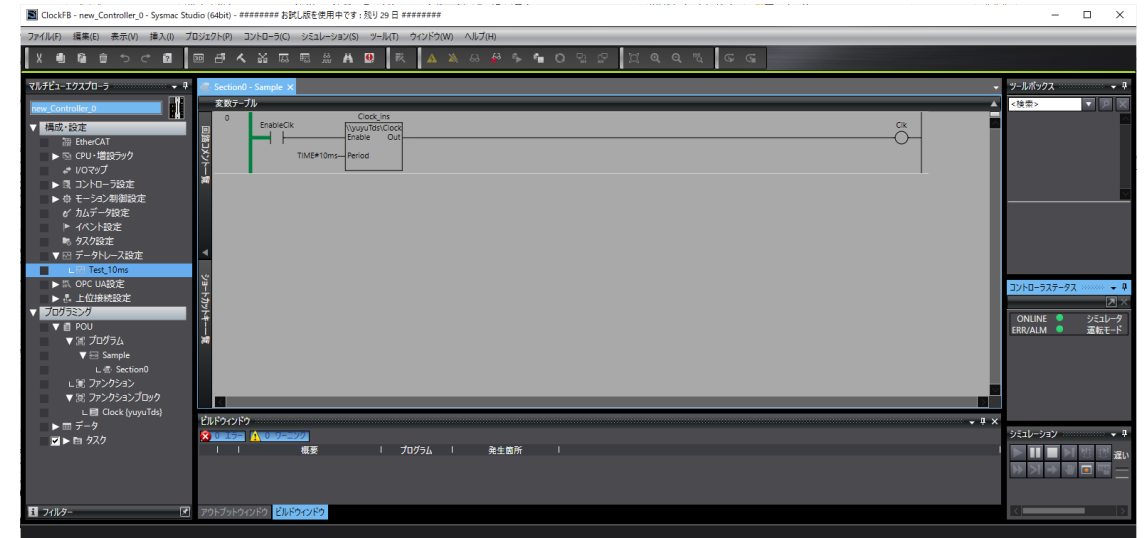
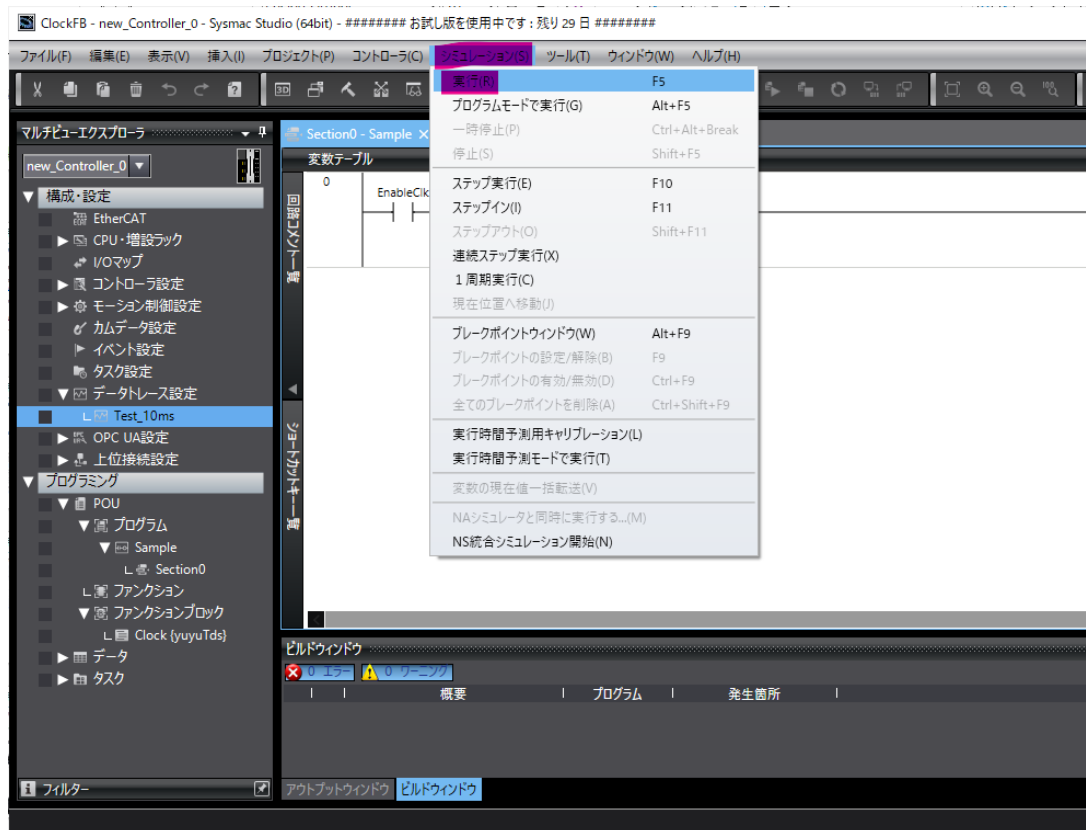
1. ダウンロードしたい「ClockFB.csm2」をダブルクリックして、SysmacStudioを起動させます。
2. 起動したら右のマルチビューエクスプローラから[プログラミング] > [POU] > [Sample] > [Section0]の順で開いて、ラダーを表示させます。



# [Appendix] ClockFBの使用手法

3. 上タブの[シミュレーション] > [実行]をクリックします。

しばらくシミュレーションが開始され、右図のようになります。（シミュレーション実行中）

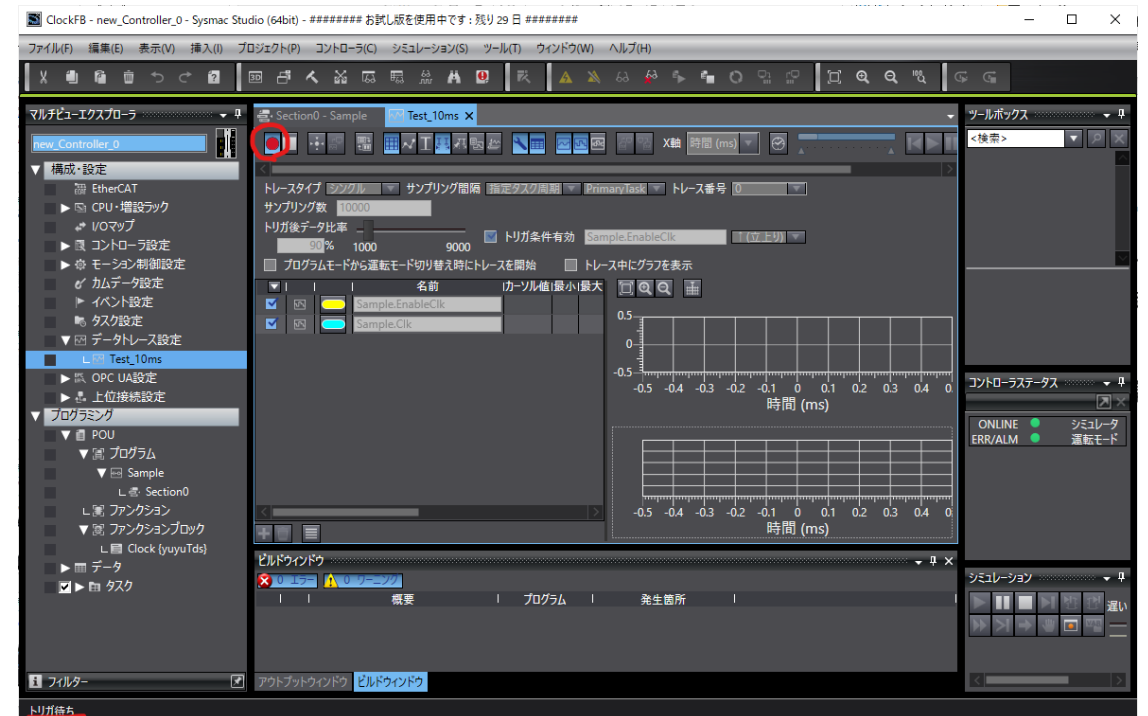
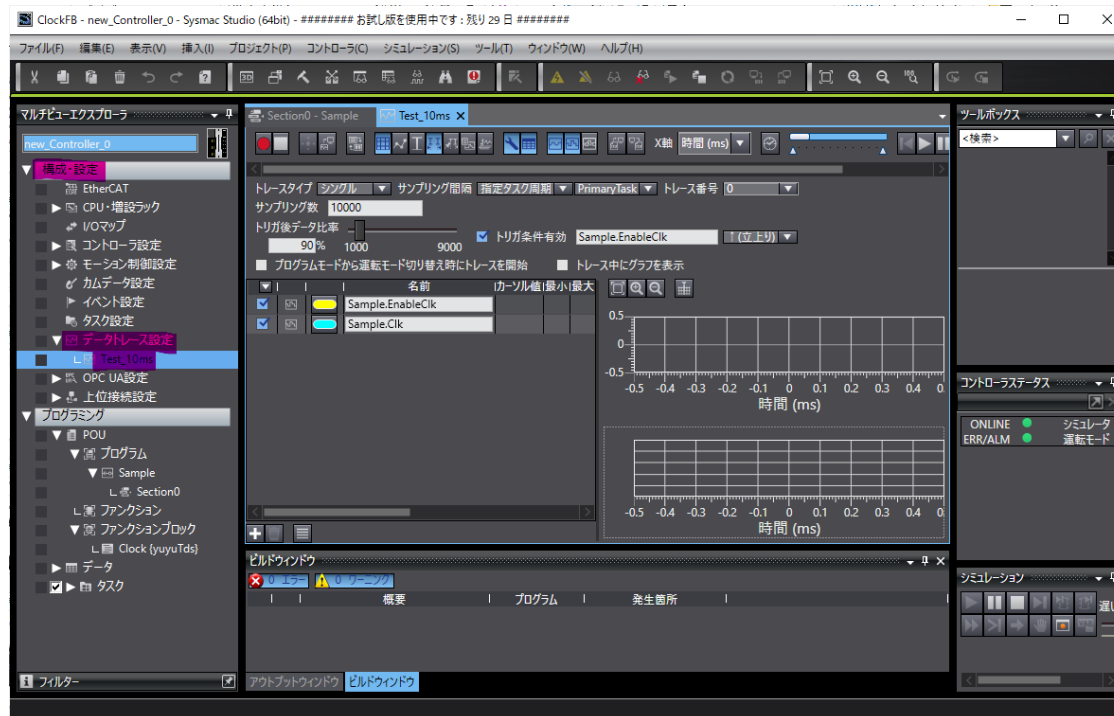


# [Appendix] ClockFBの使用手法

4. シミュレーション中の動作結果を取得するために、データトレースを行います。

右のマルチビューエクスプローラから[構成・設定] > [データトレース設定] > [Test\_10ms]を開いて、データトレース画面を開きます。

画面を開いたら、録画ボタンをクリックします。（右図のようになれば準備完了です。）



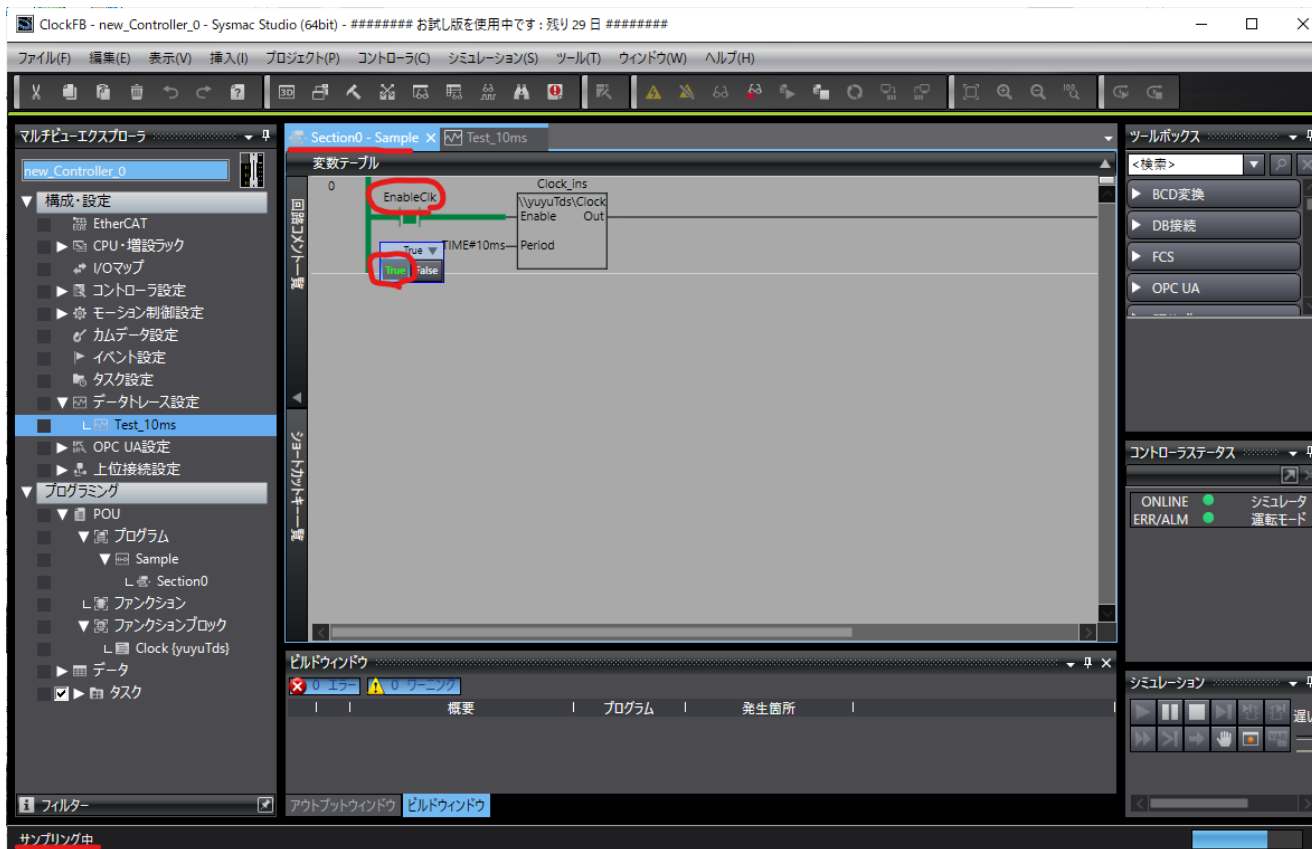
↑「トリガ待ち」と表示されている

# [Appendix] ClockFBの使用方法

5. ラダー画面に戻って、FBを実行します。

[EnableClk]をダブルクリックして、接点をTRUEにします。

FBが実行され、データトレースも自動的に開始されます。



# [Appendix] ClockFBの使用手法

6. データトレース画面に戻って、トレース結果を確認します。

10msごとのクロック信号が出力されていることがわかると思います。

(マウスホイールで波形の拡大ができます)

