

PCによるモータ制御では、通常、複数モータを高精度(1μsec以下)に全体システムとして同期制御することはできません。今回紹介するKSJシステムでは、EtherCATmasterが高精度の時計を持ち、PCの時刻ズレを吸収することにより、PCで計算したCiA402準拠のモーション指令に従いながら、EtherCAT DC機能により、複数モータを1μsec以内の同期精度のシステムを安価に構築する事が可能で、当社のシステムで機器固有の機能以外は非常に短期間に実現可能です。

## PC機能

(a) EtherCAT KSJ config tool  
EtherCAT masterの配下に接続される slaveの情報を設定でき、データ構造とパラメータ値を確認できる。

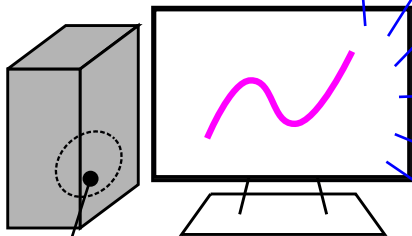
(b) 軌道生成機能  
PCツール上で、モータ動作の位置/動作速度などを設定する。  
①ティーチングデータ収集  
②座標指定 3次元シミュレータGAZEBOを流用 or Gコード(工作機械向け位置決めコード)利用

(c) 軌道データ補間機能  
EtherCATラインに接続されている各モータに対して、EtherCAT通信周期毎(250μsecなど)のデータで補完※  
※補完とは、加減速のデータ補完、S字/台形補間など、設定される軌道/メカによって、補間内容が選択できる。

(d) データ通信機能  
補完結果のデータをPCIe or TCP/IP経由で伝達

(e) モニタ機能 EtherCAT制御側の状態を表示

(f) イベント登録機能 エラーイベントの動作設定



a)からf)の各アプリ間はAPIによってアプリ間通信をおこなっており、複雑な制御タイミングを意識しないで、機器固有の機能を実現する事が可能なシステムとなっています。

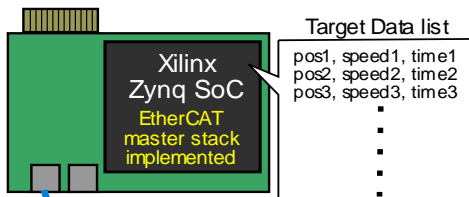


- ・PCから、目標位置/目標速度を100msec毎に伝達
- ・EtherCAT masterからは、最速で制御状態を伝達

## EtherCAT master

### Interface with PC

- ・PCIe
- ・TCP/IP (1つのPCに、複数台の EtherCATmasterが接続可能)



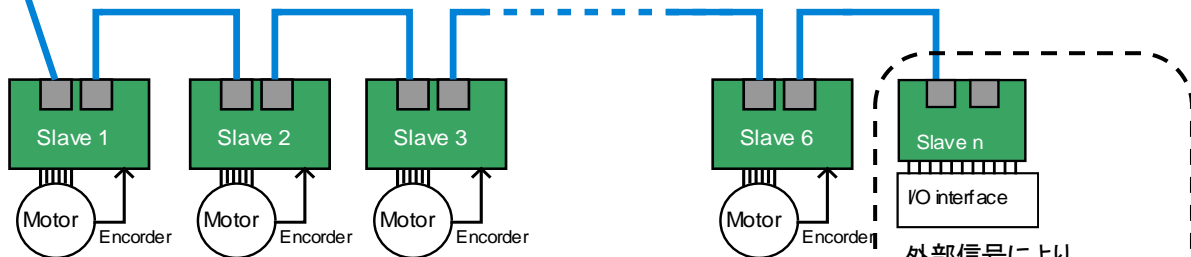
### 本基板の機能/特徴

- ①PCとの通信(PCIe or TCP/IP)
- ②EtherCAT初期化処理/CiA402プロファイル処理
- ③EtherCAT通信は、CPUと独立してFPGAで高速/高精度に制御  
DPRAM経由で、ARM CPUとFPGA間で通信データが授受可能
- ④IOユニットからのイベント制御

### EtherCATラインへの送付データについて

- ・100msec毎にPCより送られる補完データを、左図のTarget Data listのようにデータ蓄積する。
- ・PCと送受信するデータは、PC時刻精度が低い為、タイムスタンプ管理し、常に300msec分のデータバッファを保持できる仕組みを持つ。

EtherCAT cyclic time:  
250μsec (example)



- ・接続されるモータドライバ(EtherCAT slave)は、CiA402に準拠しているモータであれば、PP, Homing, CSP, CSV, CST Modelに対応可能。
- ・250μsec周期通信の場合、6個のモータドライバを同時制御可能。

外部信号により、全モータ停止、原点復帰など、予め決めた挙動を制御

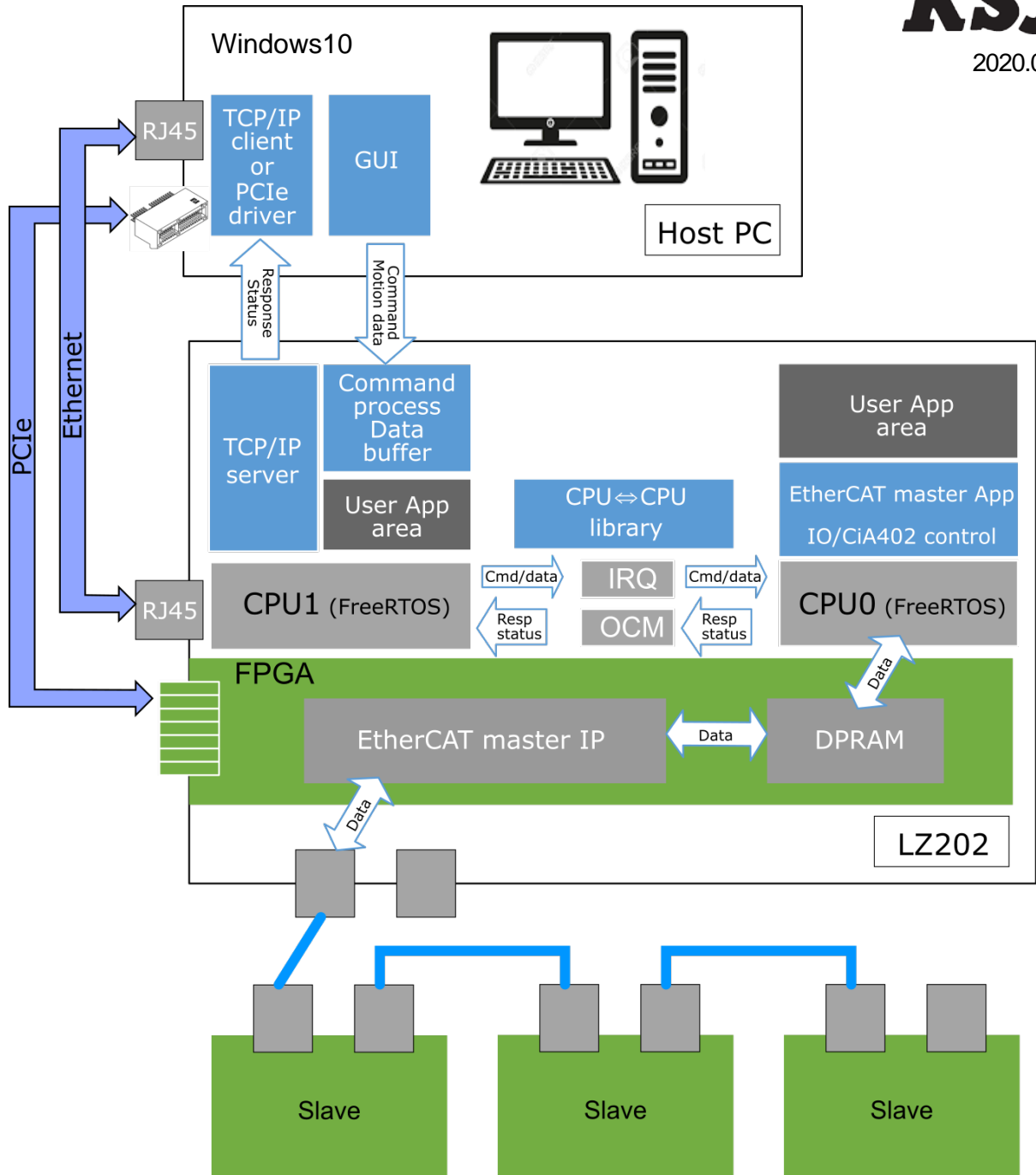
Event option

# 補足資料1: Block Diagram for motion system

モーションシステムの機能ブロック概要図について、以下に記載します。

# KSJ

2020.09



## Data sequence

1. PC send the ECAT bus scan command to LZ202 and scan all connected slaves (send scan command and call ECAT initialization API)
2. Generate/Prepare motion parameters and motion data (position, velocity, torque etc)
3. Set motion parameters and power on motors (send parameter set command and call ECAT SDO API, send CiA402 power on command and call CiA402 sequence)
4. Send Ethercat cycle based motion data and run motion in each 100ms (send motion command and divide and send to slaves)
5. User applications can be implemented using OS API, EtherCAT API, Communication library API

# 補足資料2: PCソフト説明 for motion system

モーションシステムに関連する各PCアプリケーション内容詳細について、以下に説明します。

## (a) KSJ EtherCAT config tool

EtherCAT masterの配下に接続される各slaveが持つデータオブジェクトに対し、EtherCATmasterを介してSDO通信でパラメータ設定が可能。

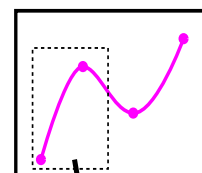
また、EtherCATmasterは配下のslaveの情報を収集でき、データ構造/パラメータ値の設定/確認が行えます。  
※機能詳細は、本ツールのマニュアルを参照ください。

## (b) 軌道生成機能

モータをどのような軌道(ルート)で動かすかを決定するために、軌道上の各位置、各速度、各トルク等を設定して、軌道生成する機能です。その軌道が実現できるかどうかは、メカ的环境や、動作させる各モータの性能を考慮して、ソフトとして判断する必要があります。

そのため、各メカ的环境に応じて、軌道生成ソフトを個別に作成する必要があり、ユーザで使用するモータ/メカ環境の情報に基づき、ソフト作成する必要があります。

※基本的にはユーザ側で作成する事が前提のアプリで、アプリによって各モータへ与える移動量や移動速度を作成すれば複雑な手順やタイミングを考慮しないでAPIにデータを書き込みc)項の「軌道データ補間アプリ」へのデータ受渡しをすることができます。

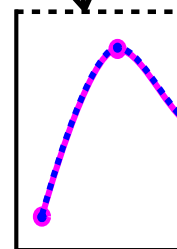


## (c) 軌道データ補間機能

上記b)の機能により生成された各軌道を基に、非常に高精度に同期制御するためには制御システムとして必要な最小時間単位で移動量、移動時間、トルク量に置き換える必要があります。このアプリの機能は最小制御サイクルごとの移動量や移動速度、トルク量を求めるための機能です。

このとき、軌道の補間(分割)方法として、線形補間、S字曲線補間、台形プロファイル補間等をユーザが選択し、データに反映する事もできます。

例えば、EtherCATmasterが各slaveに制御指令を与える周期を250μsecサイクルとすると、b)の機能で生成した軌道を、100msecを1ブロックとし、その中に250μsec単位のデータとして分割する機能を持ち、その結果をAPIに展開します。



## (d) データ通信機能

c)の機能により生成された軌道の100msecごとの補間データは、PCからEtherCATmasterに送付します。また、EtherCATmasterからは、slaveの動作状態のデータが最速でPCに送られます。

ここで、PCの時計精度は低いため、制御サイクル時間の管理はEtherCATmasterでおこない、時間の同時性を担保してるが、時計誤差によるデータ欠損を防ぐため、データバッファとして300msec分のデータを保持します。300msec分のデータバッファによって、PC時計精度が悪くても、データ欠損を予防しています。

※時刻ズレによりデータバッファ量が300msec分から乖離した時点で、必要量のデータを要求する仕組みを持ちます。

## (e) モニタ機能

本機能では、EtherCAT通信のslave側の状態をPCで収集し、表示用データとして保存する前提です。実際の画面表示には、保存している表示用データをもとにして、ユーザ側で扱いやすい表示内容に変換処理してもらうことを想定しています。

## (f) イベント設定機能

EtherCATslave(motor)を制御している際に、何かしらの異常/イベントが検知された場合に対応するため、イベント対応として、予め決められた動作をするように設定できる仕組みを提供します。

(予め決められた動作とは: 緊急停止、原点復帰、など)

このイベント対応は、EtherCATラインにおいて、画像処理スレーブが処理結果を返した場合や、IOユニットへの入力起きた場合、モータ異常を検出した場合、PCとの通信に失敗した場合、などを想定しています。

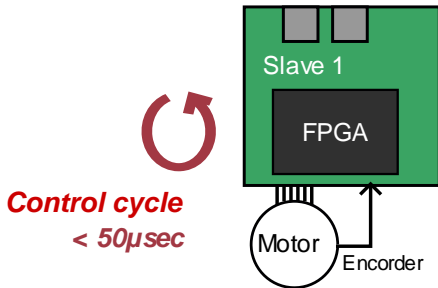
これらの場合には、PCからの指令を無視して、予め設定されている制御内容へ対応します。EtherCAT側だけで判断するため、高速に処理できる特徴を持っています。

# 補足資料3: KSJ Slaves for motion system

KSJで開発したSlave基板のうち、モーションシステムに適した各Slaveについて御紹介します。

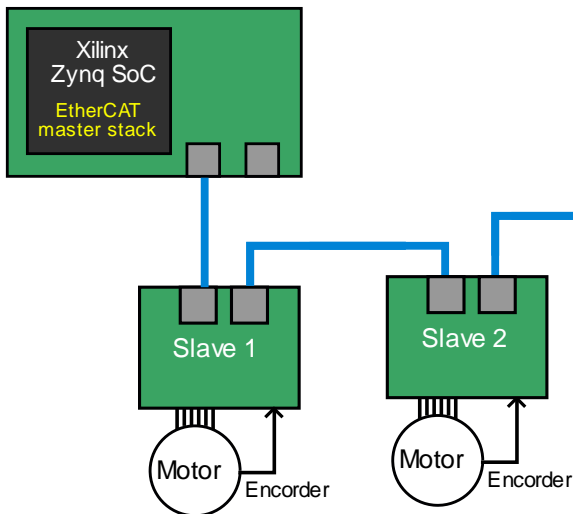
## 1. KSJ servo motor driver (Slave)

### High speed control cycle by FPGA



エンコーダ入力、フィードバック処理、モータ出力までをFPGA処理によって実現しているために、非常に高速なDCサーボドライブ基板を実現しています。これにエンコーダ付きDCブラシレスモータをEtherCAT通信(CiA402)による指令によって、速度、トルク(電流)、位置決め制御、を実現しております。  
当社のシステムでは、EtherCAT slave機能をFPGAで実現しているため、制御指令はもちろんのこと、リアルタイムで制御状態の情報を確認することが可能となっています。  
当社が提供する基板をお使いいただく事も可能ですが、御社専用の基板を作ることによって御社独自の非常に安価にDCサーボモータAmp基板を構築する事ができます。

## 2. KSJ image process unit

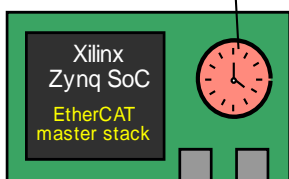


### Image processing unit

FPGA処理により、高速かつ正確なタイミングで、処理結果が取得可能。  
グローバルシャッターを用いたCMOSセンサは、動体物の画像取得も可能。  
また3次元解析のアルゴリズムも確立済みで、組み込んで使用する事ができます。  
画像処理結果からモータ等の移動量を求め、その結果をモータSlaveに通知し、モータの移動補正量などに使用する事が可能です。  
同時にモーションシステムのイベントトリガーとしても有効利用可能な画像処理ユニットです。

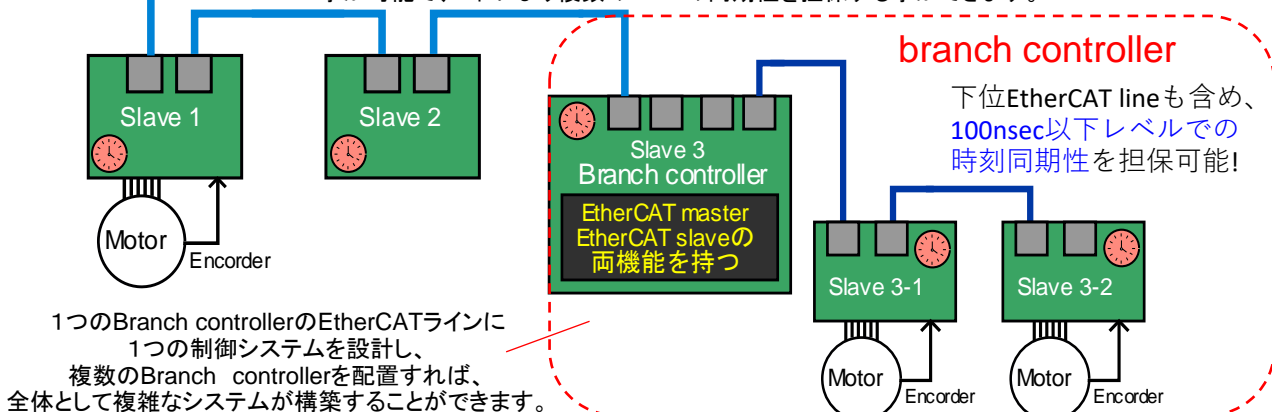
## 3. KSJ branch controller & clock synchronization system

### Reference clock for EtherCAT system



通常のEtherCATの時刻同期機能(DC機能)では、EtherCATラインに対し、1番目のスレーブの時計を、リファレンスクロックとして配信するのが一般的です。これはMasterをPCで構築する前提としているため、1番目のSlaveの時計を使うしかないので、この方法を採用しております。  
しかし、KSJのシステムでは、MasterのEtherCAT制御をFPGAで構築しているため、Masterの基準クロックをリファレンスクロックとして、各スレーブが同期がとれる技術確立しました。これによりMasterも含む全てのシステム内の時間の同時性は担保されます。

ここでSlaveの接続数が少ない場合は、高速の通信サイクルとその同期性は問題ないが、Slaveの接続数が多い場合は高速の通信サイクルが確保できなくなります。そこで下図のように、Slave機能&Master機能を同時に持つ、2階層のEtherCATラインシステムを構築し、Slaveの接続数が多い場合でもシステムの高速度と同期性が担保できる「Branch controller」を提供する事が可能で、これにより複数のSlaveの同期性を担保する事ができます。



1つのBranch controllerのEtherCATラインに1つの制御システムを設計し、複数のBranch controllerを配置すれば、全体として複雑なシステムが構築することができます。