

江端さんのDIY奮闘記 EtherCATでホームセキュリティシステムを作る(2):

EtherCAT通信の仕組みを知ろう～メイドは超一流のスナイパー!?

<http://eetimes.jp/ee/articles/1505/29/news028.html>

今回は、EtherCATの仕組みを信号レベルでご説明します。「ご主人様(EtherCATマスタ)」と「メイド(EtherCATスレーブ)たち」が、何をどのようにやり取りをしているのかを見てみると、「メイドたち」が某有名マンガのスナイパーも腰を抜かすほどの“射撃技術”を持っていることが分かります。後半では、SOEM(Simple Open EtherCAT Master)を使ったEtherCATマスタの作り方と、簡単なEtherCATの動作チェックの方法を紹介しましょう。

2015年05月29日 09時00分 更新

[江端智一, EE Times Japan]



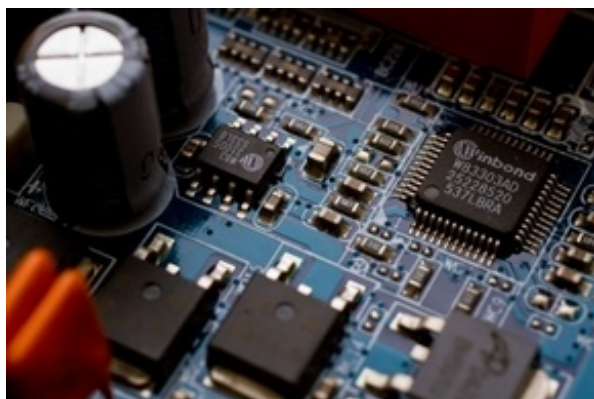
FA(ファクトリオートメーション)を支える「EtherCAT」。この超高度なネットワークを、無謀にも個人の“ホームセキュリティシステム”向けに応用するプロジェクトに挑みます……!! ⇒[「江端さんのDIY奮闘記 EtherCATでホームセキュリティシステムを作る」連載一覧](#)

私が大学生の頃、パーソナルコンピューター、通称「パソコン」を、使いこなすことができる者は、ほんの一部の人だけでした。現実には、その当時の9割のパソコンは、新品のまま、押し入れの中で眠っていたそうです。

パソコンを使いこなす者たちは、社会から隔離され、自室や研究室のスミに追いやられ、パソコン通信を行う者に至っては、「オタク」の称号を付され、「ネクラ」と決めつけられ、さらには、(少女殺害事件などと結び付けられて)潜在性犯罪者とまでうわさされる——そういう陰湿な時代でした。

その一方で、パソコンを使うだけでは足りず、パソコンのハードウェアの改造に乗り出すという豪の者もありました*)。

その者は、パソコンの筐体を開き、電子回路が搭載されているマザーボードに得体のしれない部品を差し込み、そこからケーブルの束を引き出して、自作のロボットや家電製品に繋ぎ、さらには、電話とモデムを使って、それらを遠隔からコントロールしようとしていました。



画像はイメージです

私です。

＊) 正確に言うと、そういう凄いスキルをもった友人が、当時の私にはいたのです。

□

こんにちは。江端智一です。

大学生の頃、私がやろうとしていたことは、コンピュータとネットワークによるモノの制御、今でいう、IoT (Internet of Things) の走りだったと思います。

最近、「IoT」が騒がれていますが、そもそも、コンピュータによるモノの制御なんて、別段新しいものではありません。新幹線、信号機、発電所、配電、配水、その他の社会インフラシステムのほとんどは、半世紀も前からコンピュータ(メインフレーム)が動かしてきました。

今、(というか、今さら)「IoT」が騒がれる理由は、「モノ」が、「インターネット」という誰でも使えるネットワークにつながるという点にあります。

IoTには、「オープン」というメリットがあります。

第一に、同一の通信方式やオブジェクトモデルを使うので、安価に、装置やシステムが作ることが可能となります。

第二に、制御情報の公開や共有も可能となります。

例えば、信号機がインターネットに接続されて、信号が青になるタイミングをカーナビなどであらかじめ知ることができたら、「一度も信号に止められずに運転する」というナビゲーションが(理屈としては)可能となります。

一方、IoTには「セキュリティ」という問題もあります。

(万に一つもないと思いますが)原子力発電所のシステムがインターネットにつながったとしたら、あっという間にテロリストの攻撃によって、システムダウン(最悪の場合、メルトダウン)に追い込まれることでしょう。

無限数のDIOが手に入る

EtherCATも同じスレッドで語ることができます。

「オープン」については、イーサネット(Ethernet)ケーブルという、誰にでも簡単に手に入るケーブルとパソコンを使って、100マイクロ秒単位でロボットや製造ラインを制御することができます。

「セキュリティ」については、もし、その制御システムをコントロールするパソコンが、インターネ

ットなどと接続可能な状態であれば、そのパソコンを乗っ取られ、最終的には、(工場、生産ライン、軍事基地等の)制御システムを丸ごと乗っ取られます*)。

*)『制御用のパソコンをインターネットに接続するマヌケがいるか?』という方には、[この記事の「スタックスネット」](#)をご一読ください。

—— とまあ、いろいろ書きましたが、正直なところIoTやEtherCATの、一般向けのお話は、私にはどーでも良いことなのです。

これまで、いろいろな制御システム向けのネットワーク(これを「制御LAN」といいます)に関わってきたこの私が、今回、EtherCATに狙いを定めた理由は、簡単です。

「イーサネットケーブルにつなぐだけで、事実上、無限数のDIO(Digital Input/Output)ポートが、手に入る」—— この一言に尽きます。

“DIO”とは、5V電圧と0V電圧を、それぞれON信号、OFF信号として、入出力をする信号ポートのことです。DIOポートさえあれば、(理屈の上では)どんな機械でも制御可能です。

学生の頃、私は「ロボットの歩行計画」の研究をやっていたのですが、当時の研究室のパソコン(PC-9801とかPC-286)に、たかだか8個のDIOポートを設けるために、

- 自腹を切って、I/O変換ポート用の部品を購入し、
- 拙い技量でI/O変換回路のハンダづけを試み(そして、やけどをし)、
- 友人に頭を下げて、マシン語でDIOポートのデバイスドライバを作ってもらい、
- 徹夜を繰り返しながら、制御プログラムのコーディングとデバッグを続ける

そういう日々を送っていました。

DIOポートが動き出す頃には、精根尽き果て、本来の研究(歩行ロボットの歩行計画)の気力をすっかり喪失してしまっているという、本末転倒な事態になっていました。

しかし、EtherCATなら、パソコンの筐体を開くことなく、イーサネットの口だけあれば、ほとんど無限数とっていいDIOポートが、驚くほど簡単に、安価に手に入るのです。

—— もしタイムマシンがあったら、このEtherCATスレーブを、30年前の大学の研究室に放り込んでくるのに!

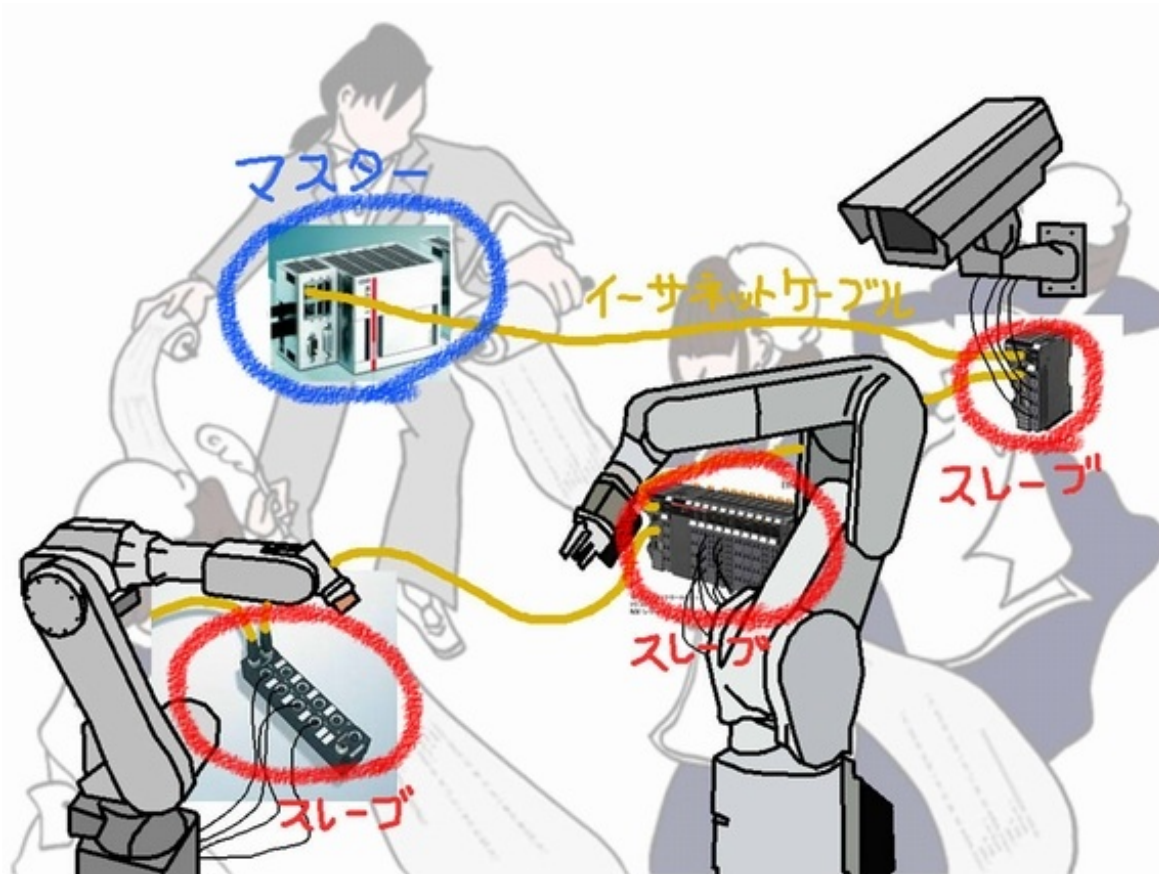
と、悔しい思いをしています。

まあ、過去のことはどうでも良いです。

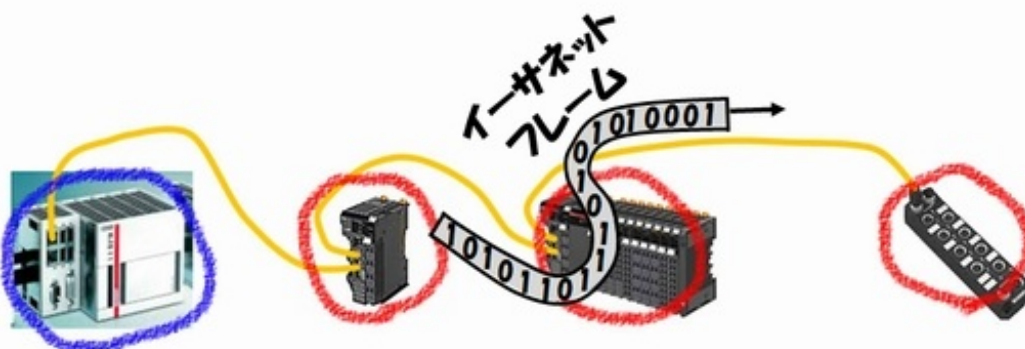
今の私には、大量のDIOをわが家の至る所に準備する、切実な理由があります。これについては、次回お話をさせてください(EtherCATと直接関係のない話を、これ以上続けると、さすがに怒られそうですので)。

「メイド」は射撃の名手!?

前回は、「ご主人様」と「メイドたち」に登場してもらって、かなり大ざっぱなEtherCATの説明を試みましたが、今回は、実際のモノで、物理レベル（信号レベル）の説明を試みます。



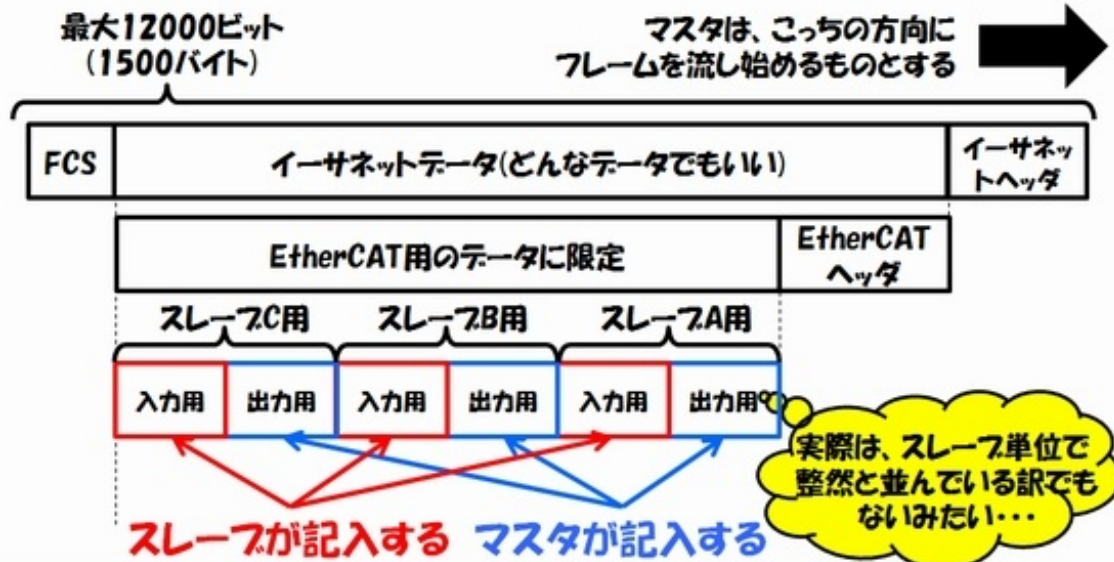
さて、このご主人様であるEtherCATマスタと、メイドであるEtherCATスレーブは、イーサネットケーブルで、「両手をつないで」ネットワークを作ります（デジチェーンといいます）。



マスタは、そのネットワークに対して、イーサネットフレームという0と1が連続するビットのかたまり（最大1万2000ビット）を送信します。このイーサネットフレームには、ネットワークにつながっている、全てのスレーブへの指示が記載されています。このイーサネットフレームは、全てのスレーブを巡回して、再度マスタに戻ってきます。

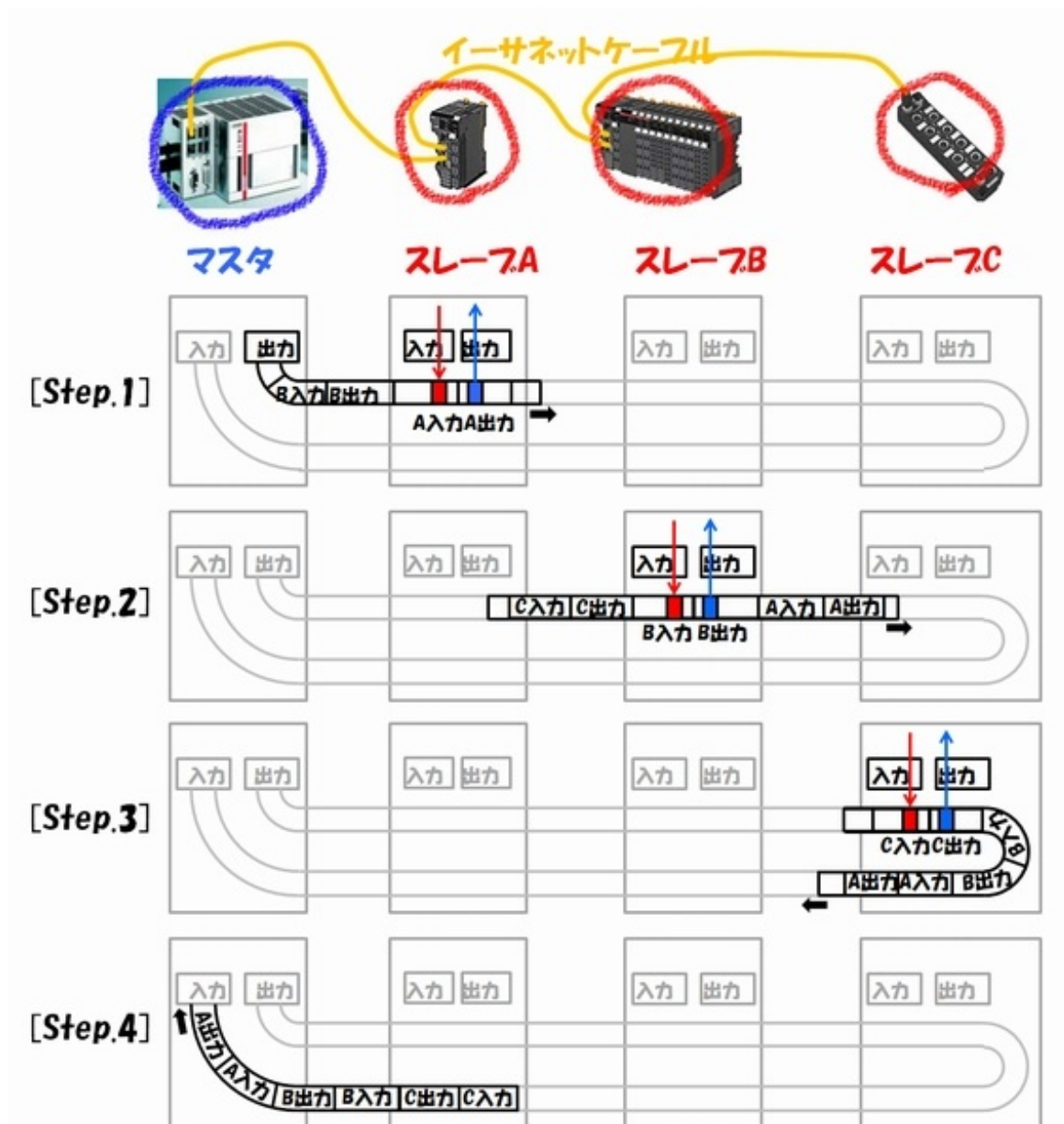
「あれ？でもこの接続(デジーチェーン)だと、ループにならないよ」とも思われるかもしれませんが、今、パソコンで使われているイーサネットケーブルは、1本のイーサネットケーブルに2本の通信路があり、「行き」と「帰り」の道が別々に存在しているので問題はありません。

イーサネットフレームの使用方法



イーサネットフレームの中に、各スレーブ用の読み書き場所が割り当てられている

上の図は、スレーブが3つ(スレーブA、B、C)があった時の、イーサネットフレームの中身のザックリした概略図です。要は、EtherCATのマスタとスレーブは、ネットワーク上で動き続けている1つのイーサネットフレームに、必要なデータを読み書きして、お互いに交換しているだけです。



上記の図は、流れている最中のイーサネットフレームから、スレーブがデータを取り出したり、書き込んだりしている様子を示す図です。

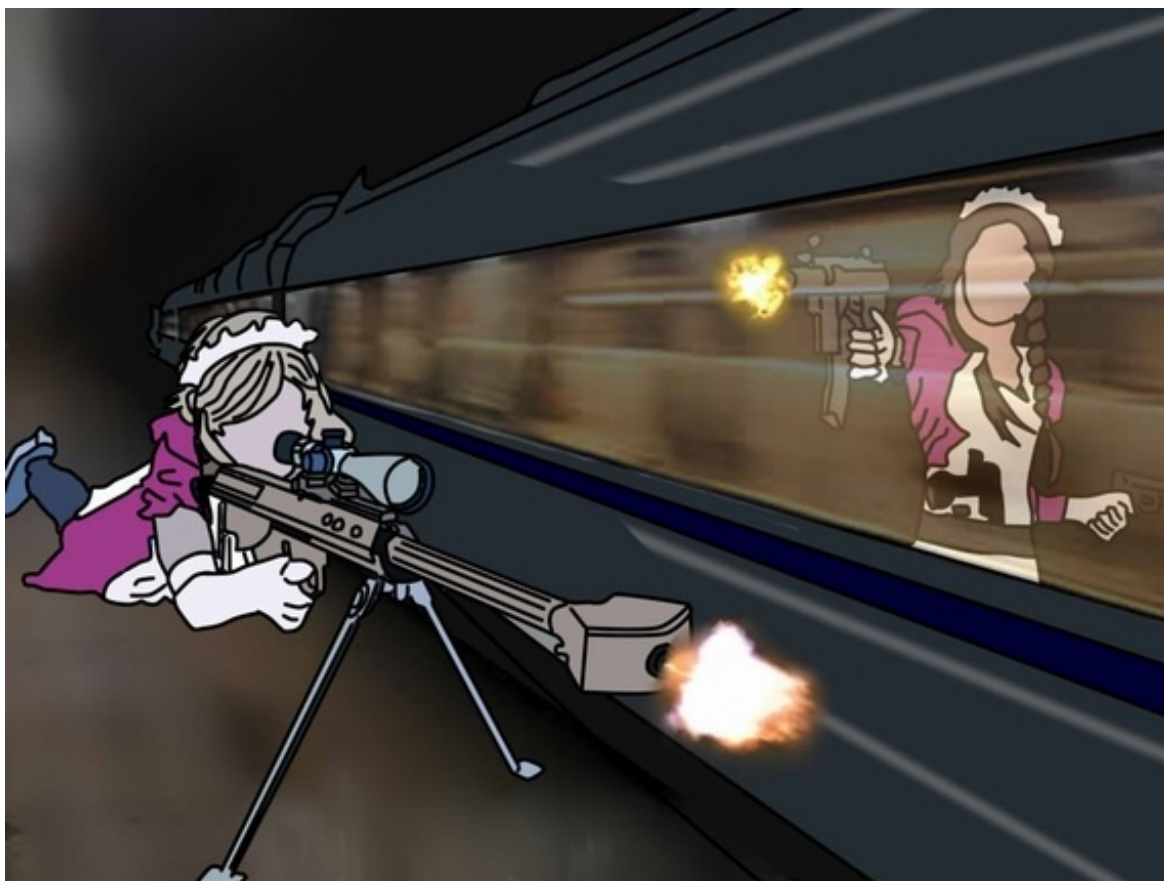
イメージとしては、

『駅を通過している最中の新幹線に、ホームから、指定された車両の座席に向けて、正確にピンポイントの狙撃をする』

または、

『駅を通過している最中の新幹線の座席から、ホームに向けて、正確にピンポイントの狙撃をする』

という感じでしょうか。しかも、連続射撃です(数十発から数百発の連射)。



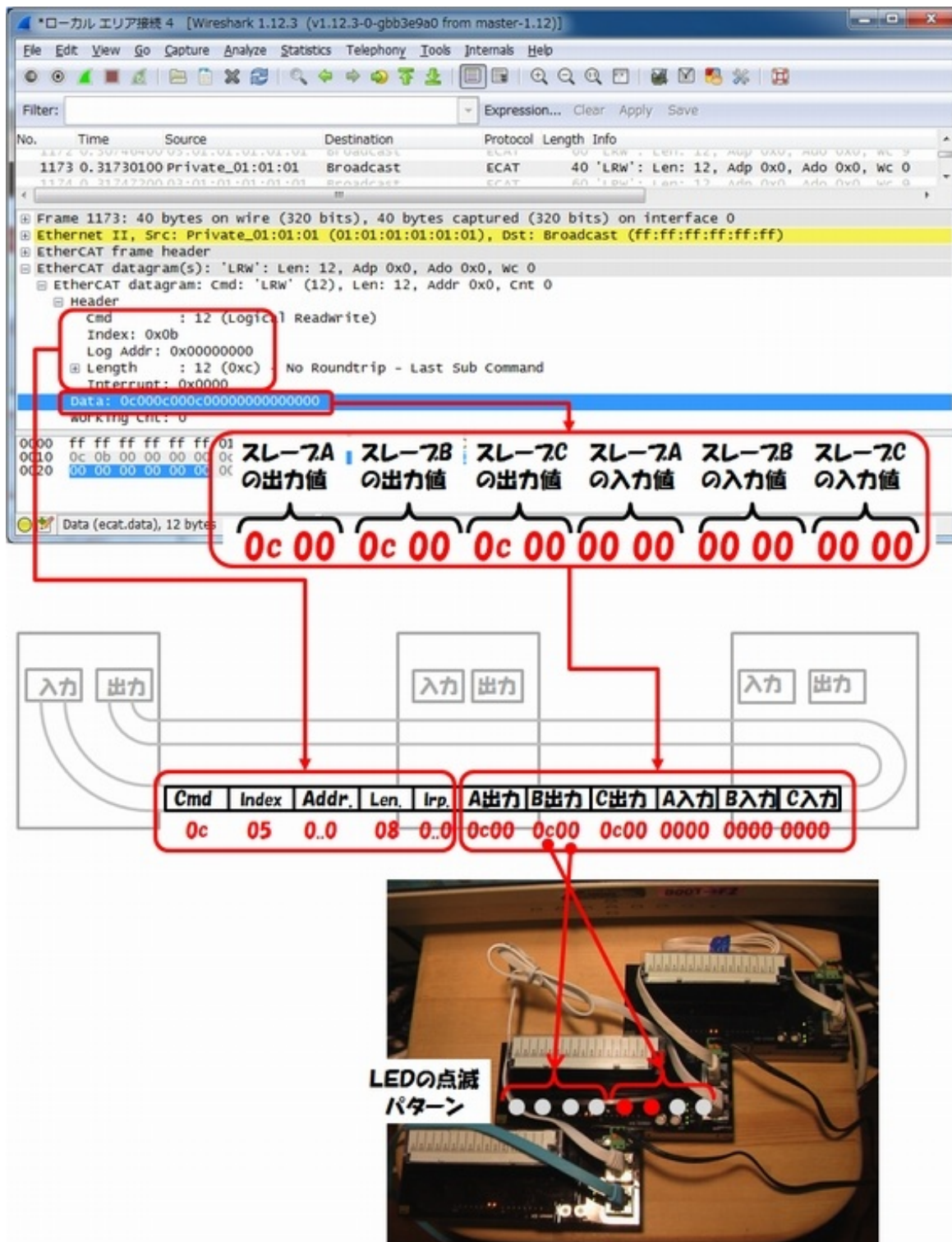
「ゴルゴ13」でも難しいだろうと思われる、高速移動中のピンポイント連続狙撃を、メイド（スレーブ）たちはやすやすとやっているわけです。

流れているイーサネットフレームを見てみる

では、EtherCATスレーブ3台（GX MD-1611）を用いて、実際に流れているイーサネットフレームを見てみましょう。

こちらが稼働中の動画になります。

下記が、ネットワークアナライザ「Wireshark」でキャプチャした、イーサネット上のフレームの内容です（一周してマスタに戻ってきたフレームです）。



このフレームは、マスタが、3つのスレーブのLEDを点滅するように指示したものになります。マスタは、3つのスレーブに対して、同じ命令「点(1) 点(1) 消(0) 消(0) 消(0) 消(0) 消(0) 消(0) = 1100 0000 = &0c00」を出しています。

今回は、4つのパターンを切り替えて、LEDを点滅させています。

このように、実際にスレーブに制御データを送って、制御システムを稼働させる通信を、プロセスデータオブジェクト(Process Data Object:PDO)通信と言います。

ちなみに、EtherCATのPDO通信は、とても「せわしない」です。マスタは、毎秒100～1000個のフレームを流し続けるのです。

例えば、上述したLEDの点滅の制御を例に取りますと、LEDを点滅させない時でも、マスタは、同じ内容のフレームを送信しています。

—— なんと、無駄なことを……。

LEDのパターンを変化させる時だけ、PDO通信すればよさそうなものです。EtherCATスレーブをつないだイーサネットのハブのLEDがすごい勢いで点滅しているのを見ると、「電気がもったいない」という気持ちになってきます。

しかし、この「せわしない」通信こそが、EtherCATを含む、制御LANの最大の強みなのです。

例えば、通信不良が発生した時（雷による瞬時停電など）や、スレーブが目標時間内に制御を完了できなかった時（過重量によるロボットアームの動作遅延など）でも、その直後に同じ制御データがやってくるので、安心です。

EtherCATのPDO通信は、TCP/IP通信とは異なり、スレーブは、フレームが到着したことを、いちいちマスタに報告したりしません。

『ワシは難しいことはよう分からん。じゃけん、不安なら100回でも、同じこと繰り返せば、ええんじゃろう？』

この、ヤクザ映画に登場する、物事を深く考えない古いタイプの親分の考え方こそが、EtherCATの超高速のリアルタイム通信と、抜群の通信信頼性を担保しているのです（そりゃ、同じことを100回行って、全て失敗するのは確かに難しいですね）。

以上が、制御データの通信、「プロセスデータオブジェクト（Process Data Object: PDO）通信」の概要になります。

メイドの素性を徹底的に調べるご主人様

前述のケースでは、マスタは、A出力、B出力、C出力、A入力、B入力、C出力の順にデータが並べているようです。このように、必ずしも、データがスレーブ単位でまとまっているわけではありません。

イーサネットフレームのどこに何のデータを割り当てるかは、マスタ側の裁量（マスタの実装）であり、原則としてオペレータやアプリケーションからはコントロールできません。

ところで、ご主人様（マスタ）は、EtherCATがPDO通信を始める前に、自分の配下にあるメイド（スレーブ）達の素性を全部調べ上げます。

名前（製品名）、出身地（製造メーカー）、得意分野（モータ、計測）、今、誰と手をつないでい

るか（デジチェーンの対向スレーブ）を、完全に調べあげた上で、各スレーブに設定情報（イーサネットフレームの何番目を使え、など）を命じることで、処理を始められるようになります。

このように、PDO通信の前に、マスタが、スレーブの設定のために行う通信を、サービスデータオブジェクト（Service Data Object: SDO）通信というのですが —— これが、気が遠くなりそうなほど面倒くさいのです。

しかし、SDO通信を突破すれば、EtherCATを『分かったような気になれます』。

今回は、たくさんのメイドたち全員が力を合わせて、制御システム全体を単一メモリ/O空間と見たてて制御システムを稼働させる、EtherCATの粋なパラダイムをご紹介しますと思います。

SOEMでEtherCATを作ってみる

では、ここからは、先月お約束したSOEM（Simple Open EtherCAT Master）を使ったEtherCATマスタの作り方（暫定版）と、一番簡単なEtherCATの動作チェックを始めたいと思います。

（1）準備していただくもの

PC

EtherCATマスタ（ご主人様）として使うため

- ノートPCでもO.K.
- 現在、仕事で使っているPCを使えます（江端はそうしています）
- 本連載では、「Windows7」OSを前提とします



イーサネット通信カード（NIC）

Eマスタとスレーブの間で、
EtherCAT通信を行うため

- インターネット用とは別に、EtherCAT専用のネットワークを作ります
- NICをPCIスロットに装着しても良いですし、面倒ならUSB-イーサネット変換アダプタでも大丈夫です
- 私はAmazonで格安購入した「Cable Matters USB 2.0」または、「LUA2-U2-ATX - バッファロー」を使っていますが、両方とも安定して稼働中です



イーサネットケーブル マスタとスレーブをつなぐため

- 100均のもので大丈夫だと思いますが、心配ならカテゴリ5e以上であることを確認してください
- クロス/ストレートどちらのケーブルでも大丈夫のようです



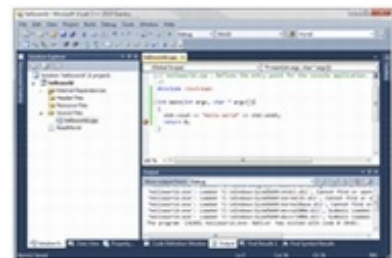
EtherCATスレーブ スレーブ(メイド)として使うため

- (なんとか入手を試みてください)
- 今回は、デジタルIOスレーブを1個だけ(オムロン社GX MD-1611)使います



Cコンパイラ、開発環境 SOEMをコンパイル/ビルドし、トレースするため

- 本連載では、[Visual C++ 2010\(無償版\)](#)を前提とします

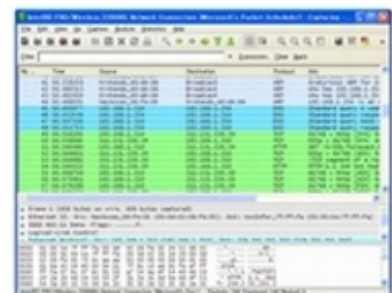


SOEMソースコード PCでEtherCATマスタを作るため

- 本連載では、[SOEM1.3.0](#)を前提とします

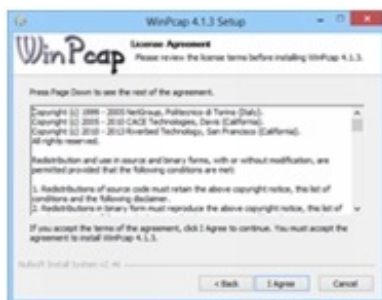
ネットワークアナライザ EtherCATのフレームを直接見るため

- 本連載では、最新のWireshark(本連載では1.12.3)を前提とします(1.12.3より以前のバージョンでは、USB-イーサネット変換アダプタのフレームがキャプチャできないようです)



パケットキャプチャライブラリ
EtherCATのフレームを直接見る
ため

- 本連載では、最新のwinpcap(本連載では4.1.3)を前提とします(最新のWiresharkをインストールすると、漏れなく最新版のインストールを尋ねてきますので、このタイミングでインストールしても良いです)



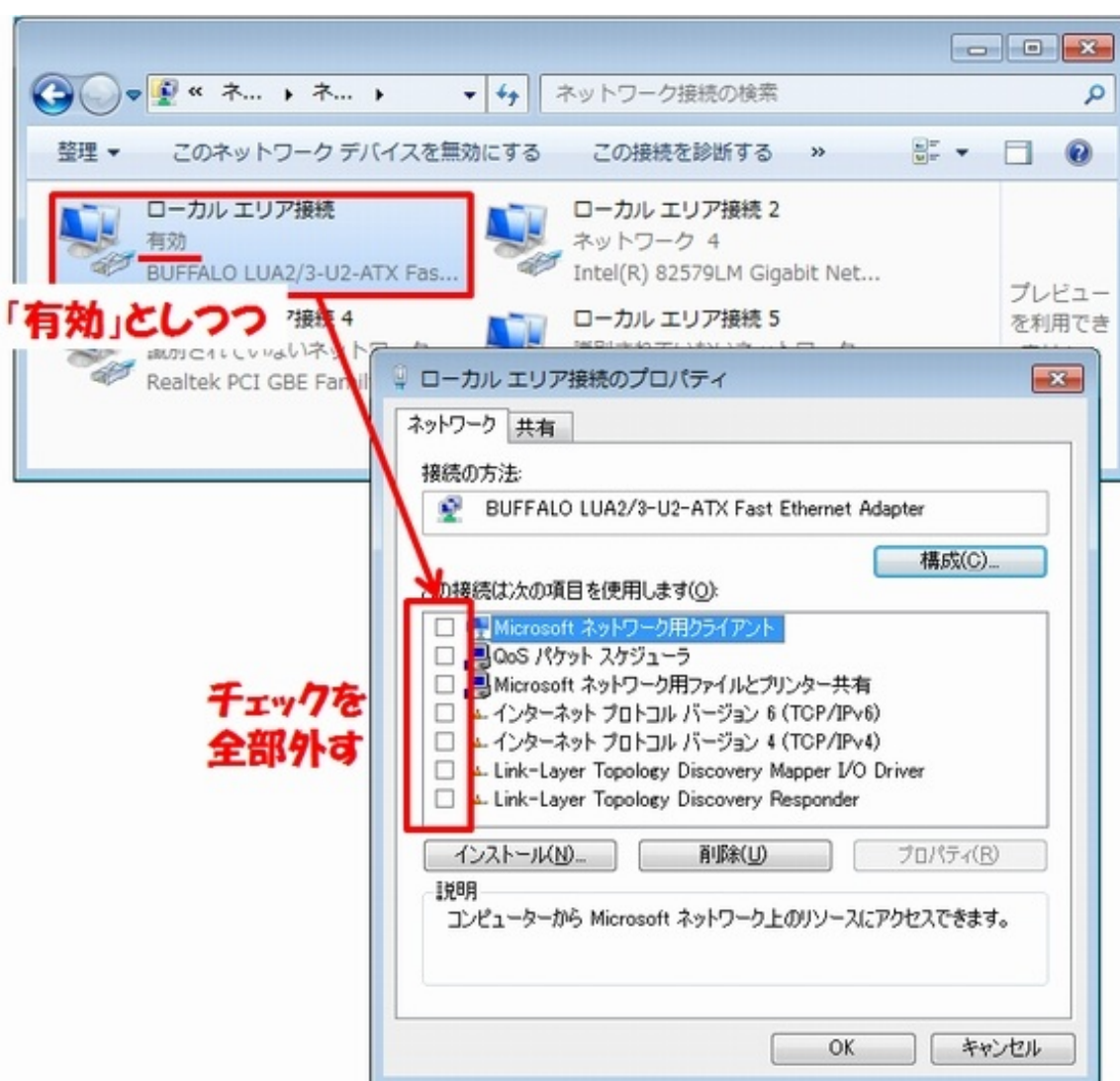
やるべきこと&確認事項

(2) やっていただくこと

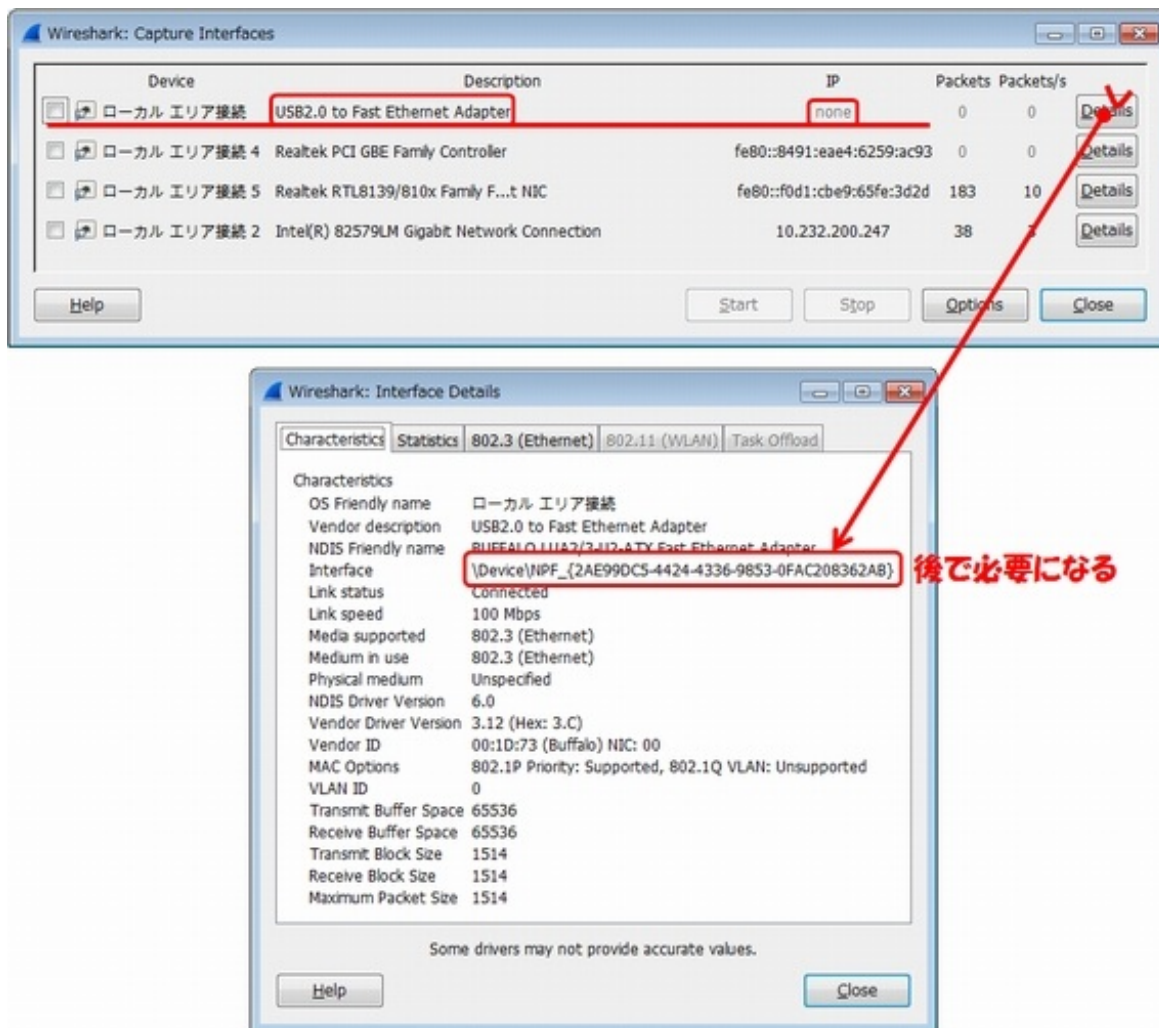
NICを増設してイーサネットケーブルでパソコンとEtherCATスレーブをつないでください。
EtherCATスレーブのRJ45ソケットは2つあるので、[IN]の方に差し込んでください。

(3) 確認していただくこと

(a) Windows7に増設したNICを、パソコンが認識していることを確認してください。適当なIPアドレスを付与しないと認識しないかもしれませんが、IPアドレスを付与したままだと、ARPやらDHCPのメッセージを撒き散らしてうっとうしいので、”192.168.0.101/255.255.255.0”などを入力して認識させた後で、全部外しておくといいでしょう(EtherCATはIPアドレスを使いません)。



WireSharkで、[Capture]→[Interface]を選択し、新設したNIC（またはUSB-NIC変換アダプタ）が見えれば、成功です。



(b) EtherCATスレーブがリンクアップをしているかを確認してください。イーサネットケーブルを差し込んだり、抜き出したりしたタイミングで、LEDが点灯、消灯すれば、リンクアップは成功していると考えて良いでしょう。

SOEMの構築

(4) SOEMの構築

SOEMのデバッグ/トレース環境構築は次回にご説明致しますので、今回は、とにかくEtherCATのスレーブを動かしてみましょう。

(Step.1) ソースコードをダウンロードする

[こちら](#)から、バージョン1.3.0をダウンロードして、適当なディレクトリに展開してください。私は、デスクトップに置きました。



(Step.2) Visual C++ 2010 (無償版) をインストールする

[こちら](#)からダウンロードできます。どこにインストールするかは、何も考えずに「OK」ボタンを押し続けていました (Cドライブになると思います)。

(Step.3) SOEMを展開したディレクトリで、以下を実行する

```
make_libsoem_lib.bat "C:¥Program Files¥Microsoft Visual Studio 10.0¥VC" x86
make_test_win32.bat "C:¥Program Files¥Microsoft Visual Studio 10.0¥VC" x86
make_test_win32_all.bat "C:¥Program Files¥Microsoft Visual Studio 10.0¥VC" x86
```

「ユニコードにしる」だのなんだのと、メッセージがやかましいですが、一応コンパイルできて、2つの実行ファイル、

```
test¥win32¥slaveinfo¥slaveinfo.exe
test¥win32¥simple_test¥simple_test.exe
```

が出来ているはずです。

もし、うまくコンパイルできなかった場合には、実際にCドライブのトップからディレクトリ名を調べて見てください。Windows7の32bitと64bitで若干ディレクトリの名称が変わっているようなので、実際のディレクトリ名に合わせてください。

(Step.4) slaveinfo.exeによる動作チェック

slaveinfo.exeは、イーサネットに繋がっているEtherCATスレーブの情報をかき集めてきて、それを表示するツールです。マスタの命令に応じて、EtherCATスレーブが、それぞれの持っているROM (EPPROM) に書き込まれた情報を、マスタに差し出すことで実現されます。

```
cd test¥win32¥slaveinfo
```

として、

```
slaveinfo.exe
```

と、打ち込むと、「NICの情報がないよー、この中から選んで付けて」と言ってくるので、

```
コマンドプロンプト
C:\Users\ebata\Desktop\SOEM1.3.0\test\win32\slaveinfo>slaveinfo.exe
SOEM (Simple Open EtherCAT Master)
Slaveinfo
Usage: slaveinfo ifname [options]
ifname = eth0 for example
Options :
  -sdo : print SDO info
  -map : print mapping
Available adapters
Description : USB2.0 to Fast Ethernet Adapter, Device to use for wpcap: %Device%\NPF_{2AF99DC5-4424-4336-9853-0FAC208362AB}
Description : Realtek PCI GBE Family Controller, Device to use for wpcap: %Device%\NPF_{36E96367-22C1-425E-95AD-6935962AAD8B}
Description : Realtek RTL8139/810x Family Fast Ethernet NIC
                , Device to use for wpcap: %Device%\NPF_{109FFFBC-99F9-4531-B69C-B28156E95B04}
Description : Intel(R) 82579LM Gigabit Network Connection, Device to use for wpcap: %Device%\NPF_{280860B0-0E41-4997-8A5B-4AC8BF215705}
End program
C:\Users\ebata\Desktop\SOEM1.3.0\test\win32\slaveinfo>
```

NIC情報がないよ～

この中から選びな～

(さっきWiresharkにあったのは)これでしょ？

先程Wiresharkで調べたNICの情報を添付します。


```
コマンド プロンプト
C:\Users\ebata\Desktop\SOEM1.3.0\test\win32\slaveinfo>slaveinfo.exe %Device%\NPF_{2AE99DC5-4424-4336-9853-0FAC208362AB}
SOEM (Simple Open EtherCAT Master)
Slaveinfo
Starting slaveinfo
ec_init on %Device%\NPF_{2AE99DC5-4424-4336-9853-0FAC208362AB} succeeded.
wkc = 0
1 slaves found and configured.
Calculated workcounter 3

Slave:1
Name:GX-MD1611
Output size: 16bits
Input size: 16bits
State: 4
Delay: 0[ns]
Has DC: 1
DCParentport:0
Activeports:1.0.0.0
Configured address: 1001
Man: 00000083 ID: 00000032 Rev: 00010000
SM0 A:1800 L: 512 F:00010026 Type:1
SM1 A:1c00 L: 512 F:00010022 Type:2
SM2 A:1100 L: 2 F:00010064 Type:3
SM3 A:1200 L: 2 F:00010020 Type:4
FMMU0 Ls:00000000 LI: 2 Lsb:0 Leb:7 Ps:1100 Psb:0 Ty:02 Act:01
FMMU1 Ls:00000002 LI: 2 Lsb:0 Leb:7 Ps:1200 Psb:0 Ty:01 Act:01
FMMUfunc 0:1 1:0 2:0 3:0
MBX length wr: 512 rd: 512 MBX protocols : 04
CoE details: 23 FoE details: 00 EoE details: 00 SoE details: 00
Ebus current: 0[mA]
only LRD/LWR:0
End slaveinfo, close socket
End program
```

**EtherCATスレーブの
構成情報(とても重要)**

このようにスレーブの構成情報が表示されます。この情報の読み方は、次回御説明致します(とても重要です)。

simple_test.exeによる動作チェック

(Step.5) simple_test.exeによる動作チェック

さて、SOEMとEtherCATスレーブが正常に動いているかどうかを、確認します。

現時点では、デジタルIOのスレーブ1台のみがつながっている状態にしてください。起動の状態を、先月お見せしたものと同じで恐縮ですが、動画(Youtube)でご覧ください。

スレーブによっては、ウンともスンとも言わない場合もありますが、今回はそれでも構いません。次回は、皆さんの環境に合わせて、simple_test.cを修正して、この「動かないスレーブ」を動かします。



先月のコラムを読み直してみると、「今回は、私がこのEtherCATの連載をEE Times Japan編集部へ提案させていただいた経緯や、また、SOEMの開発者の方とのやりとりについても、お話ししたいと思います」と書いてありますが、今回一言も言及できませんでした。

誠に申し訳ありませんでした。

今回の原稿を書き出したら、「オタク」だの「ネクラ」だのと決め付けられたキャンパスライフがフラッシュバックしてきまして、怒りにまかせて書き続けていたら、導入部だけで10ページを超える惨事になってしまいました。

慌てて削除したものの――結局、今回も長文を製造してしまいました。

私のコラムは長いので、嫁さんや娘たちは、査読後に、ゲッソリして机に突っ伏していることがあります。

担当者さんのメールの文面からは、「江端さんは、(文量を)気にしないでいいんですよ～～」と言いながら、笑顔が引きつっている様子が読みとれます。他のWeb媒体の編集担当者の方からも、打ち合わせの場で「長い。なんとかしろ」という趣旨のことを言われたこともあります。

そんな査読者、編集部そして読者の皆さんに対し、最大級の謝罪を込めて、私から申し上げることは、この一言に尽きます。

―― 諦めてください。

「英語に愛されないエンジニア」が、[何をしても英語に愛されない](#)ように、何をしても「文章を簡潔に書けないエンジニア」―― も、いるのです。

では、次回、またお会いしましょう。

特別協力:

本連載では、スレーブの提供などで[ベックhoffオートメーション](#)にご協力いただいております。



Profile

江端智一（えばた ともいち）

日本の大手総合電機メーカーの主任研究員。1991年に入社。「サンマとサバ」を2種類のセンサーだけで判別するという電子レンジの食品自動判別アルゴリズムの発明を皮切りに、エンジン制御からネットワーク監視、無線ネットワーク、屋内GPS、鉄道システムまで幅広い分野の研究開発に携わる。

意外な視点から繰り出される特許発明には定評が高く、特許権に関して強いこだわりを持つ。特に熾烈（しれつ）を極めた海外特許庁との戦いにおいて、審査官を交代させるまで戦い抜いて特許査定を奪取した話は、今なお伝説として「本人」が語り継いでいる。共同研究のために赴任した米国での2年間の生活では、会話の1割の単語だけを拾って残りの9割を推測し、相手の言っている内容を理解しないで会話を強行するという希少な能力を獲得し、凱旋帰国。

私生活においては、辛辣（しんらつ）な切り口で語られるエッセイをWebサイト「[こぼれネット](#)」で発表し続け、カルト的なファンから圧倒的な支持を得ている。また週末には、LANを敷設するために自宅の庭に穴を掘り、侵入検知センサーを設置し、24時間体制のホームセキュリティシステムを構築することを趣味としている。このシステムは現在も拡張を続けており、その完成形態は「本人」も知らない。

本連載の内容は、個人の意見および見解であり、所属する組織を代表したものではありません。

関連記事



[産業/FA/医療——新市場狙い活気づくWi-Fi市場、勝ち残る戦略は何か？](#)

今までインターネットにつながってこなかった、「非PC/モバイル」の機器が、M2Mやモノのインターネットというコンセプトと絡みつくことで、新たな市場が生まれようとしている。今後数年で競争が激しくなることが予想されるこの市場で、いかに戦っていくか。組み込み向けWi-Fiモジュールを手掛けるサイレックス・テクノロジーの戦略をまとめた。



[人類は、“ダイエットに失敗する”ようにできている](#)

今回から新シリーズとしてダイエットを取り上げます。ダイエット——。飽食の時代にあって、それは永遠の課題といっても過言ではないテーマになっています。さて、このダイエットにまつわる「数字」を読み解いていくと、実に面白い傾向と、ある1つの仮説が見えてきます。



[TOEICを斬る（後編）～“TOPIC”のススメ～](#)

今回、「TOEIC」の代替案として提唱するのが「TOPIC（Test of Playing for International Communication）」です。その名の通り、“演技をする（Play）”テストになります。つまり、自分が伝えたいことを、いかに正確にジェスチャで伝えられるかを客観的に評価するわけです。空港に行きたいとき、相手を責めるとき、トイレを我慢しているとき……。あなたはどのようなジェスチャで伝えますか？



[【NIWeek 2012】LabVIEWが奏でる“オーケストラ”が青く光る！？—数多くのデモに注目](#)



LEDアニメーションに同期させて楽器を演奏するデモやビール醸造システム、3Dプリンタなど一。2012年8月6日～9日（現地時間）に開催された「NIWeek 2012」の展示会場には、National Instrumentsのグラフィカル開発ツール「LabVIEW」や計測/制御ハードウェアを、組み込みシステム用途で活用したデモが注目を集めていた。



[リアルタイム制御とネットワーク処理を同時に実現、ルネサスのFA用マイコン](#)

ルネサス エレクトロニクスは、ハイエンドマイコン「RZファミリ」として、産業ネットワーク機能を内蔵した「RZ/T1グループ」10品種を新たに発表した。産業機器の高度なリアルタイム制御とネットワーク処理を1チップで実行することが可能となる。

Copyright© 2016 ITmedia, Inc. All Rights Reserved.

