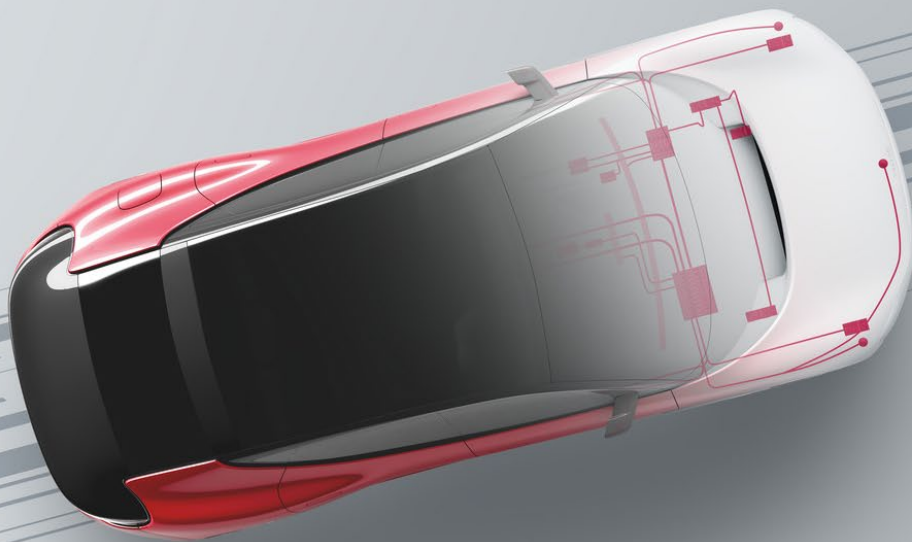


はじめての XCP

ビギナー向け資料「はじめてシリーズ」



目次

測定／キャリブレーションプロトコル XCP 入門.....	4
1. 測定／キャリブレーションプロトコルとは？	4
測定／キャリブレーションとは？	5
キャブレーターによる機械的キャリブレーション	5
ECU におけるキャリブレーション	5
機械的キャリブレーションと測定	6
ECU における測定	7
測定／キャリブレーションプロトコルへ	7
測定／キャリブレーションプロトコルの歴史	8
ASAM とは？	8
ASAM における CCP/XCP の位置付け	9
XCP のメリット	10
XCP の各ネットワーク例	11
2. XCP プロトコルの通信の仕組みと機能	12
マスター・スレーブ方式	12
ネットワークと転送方式	13
CTO/DTO	13
パケットフォーマットと PID	14
測定／キャリブレーション対象へのアクセス	16
非同期測定	16
同期測定	17
同期測定にかかわる XCP 用語と概念	18
同期測定時の処理	18
測定対象と DTO パケット	19
同期測定のオプション	20
キャリブレーション	21
その他の XCP 機能	21
3. XCP の動向と適用事例	22
測定／キャリブレーションのシステム構成	22
XCP スレーブドライバとその提供	23
総合化された測定／キャリブレーションシステム	23
XCP の応用事例	25
EV/HEV 自動車開発における XCP の利用	26
適用事例: XCP on Ethernet とベクターの VX1000 システムを使った測定	26
モデルベース開発における XCP の利用	27
適用事例: ベクターの「CANape」と「Simulink XCP Server」を使った測定／キャリブレーション	28

4. お問い合わせ窓口 29

本稿はベクター・ジャパン執筆による原稿に基づき、2011 年に@IT MONOist に連載された記事「測定／キャリアレーションプロトコル XCP 入門」より転載、加筆、修正したものです。

@IT MONOist
<http://monoist.atmarkit.co.jp>

発行元:ベクター・ジャパン株式会社

ベクター・ジャパン株式会社の書面による許可なしに、本書の内容を転載、複製、複写することを禁じます。
記述されている内容や製品画面の表示名称は予告なく変更されることがあります。

測定／キャリブレーションプロトコル XCP 入門

1. 測定／キャリブレーションプロトコルとは？

自動車をはじめとする“制御”を伴う製品開発の測定／キャリブレーション段階で使われる汎用プロトコル「XCP」を解説します。

自動車や一般産業機器のような“制御”を伴う製品開発には、「キャリブレーション」、あるいは「適合」「マッチング」と呼ばれる工程が存在します(図 1)。

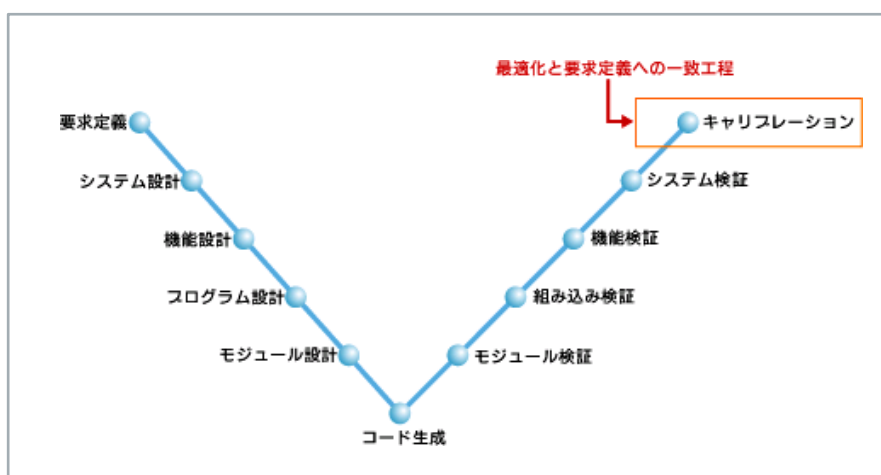


図 1: 製品開発の工程とキャリブレーション

※ベクター・ジャパンの資料を基に作成

この工程では、電子化された制御を行う「電子制御ユニット(ECU: Electronic Control Unit)」と、制御対象であるアクチュエータなどの「コントロールデバイス」や制御の指標を求めるための「センサ」とがつながった状態で、制御全体の最適化を行って要求定義に合わせる作業が行われます。具体的には、最適化を行うために ECU の制御乗数や制御要因パラメータなどを変更したり、変更した結果の確認を行うために ECU へアクセスしたりといった作業です。

この ECU へのアクセスを行うためのプロトコルが、「測定／キャリブレーションプロトコル」と呼ばれるものです。アクセスする側である測定／キャリブレーションツールとアクセスされる側である ECU との間のプロトコルを規定しておくことで、異なるマイコンや異なる物理アクセス環境であっても、同一のプロトコルを使った統一した測定／キャリブレーションが可能になります。

本稿では、測定／キャリブレーションの紹介やキャリブレーションプロトコルの必要性を説明するとともに、測定とキャリブレーションのための汎用プロトコル「XCP (Universal Calibration Protocol)」のメリットやプロトコルの説明を行い、さらにはその動向や具体的な事例としてモデルベース開発および HV (Hybrid Vehicle) / EV (Electric Vehicle) の開発における XCP の利用例について紹介していきます。

まず、「測定／キャリブレーションとは何か？」からはじめ、プロトコル化やその位置付け、メリットおよびネットワーク例について説明したいと思います。

測定／キャリブレーションとは？

そもそも“測定”とはなんでしょうか。用語の意味としては、「長さ・重さ・速さなど種々の量を器具や装置を用いて測ること」です。そして、“キャリブレーション”ですが、用語の意味としては「条件・状況などに当てはまること」になります。

まずは、測定／キャリブレーションがどのようなものなのかを理解いただくために、キャリブレーションが機械的に行われている「キャブレーター」を例に挙げ、さらに電子化されているキャリブレーションとそのプロトコル、測定について解説します。

キャブレーターによる機械的キャリブレーション

キャブレーターとは、内燃機関で“燃焼”を行わせるために、燃料と空気を混合させる装置のことで、適切な混合気を作り出す制御を行っています。この制御は、機械的に行われており、アクセルの入力に応じてキャブレーターが混合気の燃料と空気の配分を決定します。

キャブレーターには、混合気の燃料と空気の配分を決めるための「ネジ」が複数付いています。それぞれのネジにより、アイドリング時やアクセル中開、全開時の混合気の配分を決定できるほか、エンジンやマフラーなどに合わせた適切な混合気を作り出すための調整が行えます。

この調整行為そのものがキャリブレーションになります。つまり、機械的な制御装置であるキャブレーターにおいて、接続しているエンジンやマフラーなどの物理的な条件・状況に混合気を当てはめることがキャリブレーションであり、キャリブレーションを行う道具は「ネジ回し」であり、その対象はネジになります(図 2)。

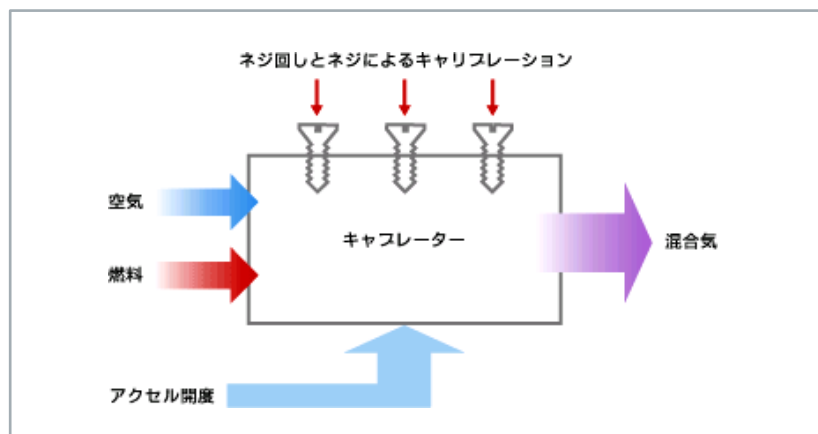


図 2: キャブレーターのキャリブレーション

※ベクター・ジャパンの資料を基に作成

ECU におけるキャリブレーション

前述したキャブレーターによる機械的な混合気制御および燃焼制御は、今日、電子化され、ECU 内のソフトウェアによる制御に置き換わっています。かつてネジが行っていたような調整部分は、すべてソフトウェア上の「パラメータ」として扱われています。

しかしながら、ソフトウェアによる制御に置き換わったとしても、物理的な条件・状況に応じたキャリブレーションが必要である点は変わりありません。そのため、キャリブレーションを行う道具、つまり、機械的なネジ回しの代わりになる道具が必要となります。このキャリブレーションを行うために ECU へアクセスする道具のことを「キャリブレーションツール」と呼びます。

さらに、キャリブレーションの道具が必要であることと同時に、“道具の統一化”が必要であることも、ネジとネジ回しの機械的キャリブレーションから考えることができます。ネジ回しが、どのような機械的制御装置でも使用できることが理想であるように、異なる ECU ソフトウェアのパラメータを同一の道具でキャリブレーションできることが求められます。

このように、機械的制御から ECU 内のソフトウェア制御に置き換えられたこと、そして、キャリブレーション工程をどのように実現すべきなのかという要求からキャリブレーションプロトコルが生まれたのです。表 1 に、機械的キャリブレーションと ECU キャリブレーションの対比を示します。

	機械的キャリブレーション	ECUキャリブレーション
対象	ネジ	ソフトウェア上のパラメータ
道具	ネジ回し	キャリブレーションツール
方法	ネジを回す	パラメータへのアクセスと変更
統一化	道具の物理的統一	プロトコルによる統一

表 1: 機械式キャリブレーションと ECU のキャリブレーションとの対比表

機械的キャリブレーションと測定

機械的制御を行っているキャブレターの場合、混合気や燃焼後の排気ガスなどをセンサで測定することでキャリブレーション結果を知ることができます。つまり、キャリブレーションを行うときには、必ず測定が必要になります。

機械的キャリブレーションにおいては、測定とキャリブレーションは 1 対であり、キャリブレーションした結果の指標を求めるために測定が必要であるといえます(図 3)。

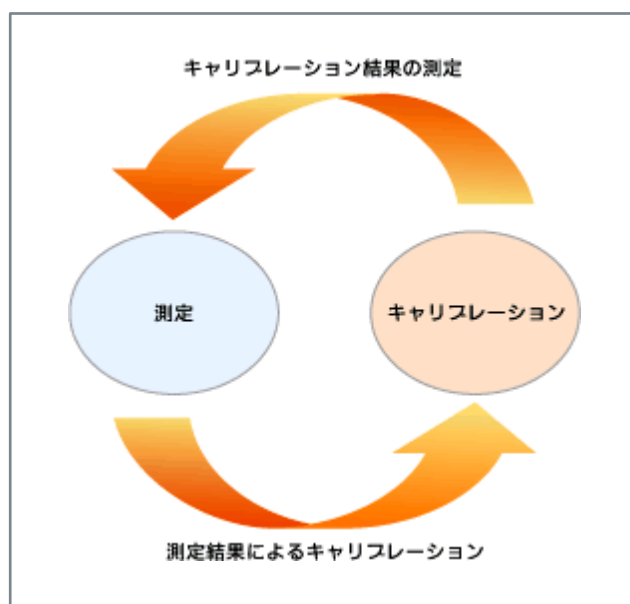


図 3: 測定とキャリブレーションの関係

※ベクター・ジャパンの資料を基に作成

ECU における測定

電子化された ECU 内のソフトウェア制御においても測定は必要ですが、要求される測定項目が追加されます。前述の機械的キャリブレーションと同じく、センサを使った最終的な制御結果の測定に加え、制御しているソフトウェアの内部値の測定が必要になります。

このソフトウェアの内部値の測定を行うことで、制御そのものの妥当性を調べることができます。さらに、ECU 内のソフトウェアでは混合気制御だけではなく、燃焼制御など、ほかの制御を組み合わせた統合的な制御を行っています。この場合、ソフトウェアの内部値である制御間の受け渡し値を測定しない限りその妥当性を知ることはできません。

測定がキャリブレーションと同じ状況で必要とされ、かつ ECU ソフトウェア内部に対するアクセスが必要とされることから、キャリブレーションツールには必ず「測定ツール」としての機能が含まれるようになり、測定もまた「プロトコル化」され、統一されることになりました。

測定／キャリブレーションプロトコルへ

測定とキャリブレーションが、同じ状況で同一対象へのアクセスを行うということから、これを実現するための機能を持つ、測定／キャリブレーションプロトコルが誕生しました。

本稿の主役である XCP とは、測定／キャリブレーションツールと ECU ソフトウェア間の通信を行い、キャリブレーションするパラメータおよび内部測定値へのアクセスを可能にするプロトコルです。また、ツールとソフトウェア間の CAN (Controller Area Network) や FlexRay、Ethernet などのあらゆる通信媒体において利用できるように汎用化されたプロトコルになっています(図 4)。

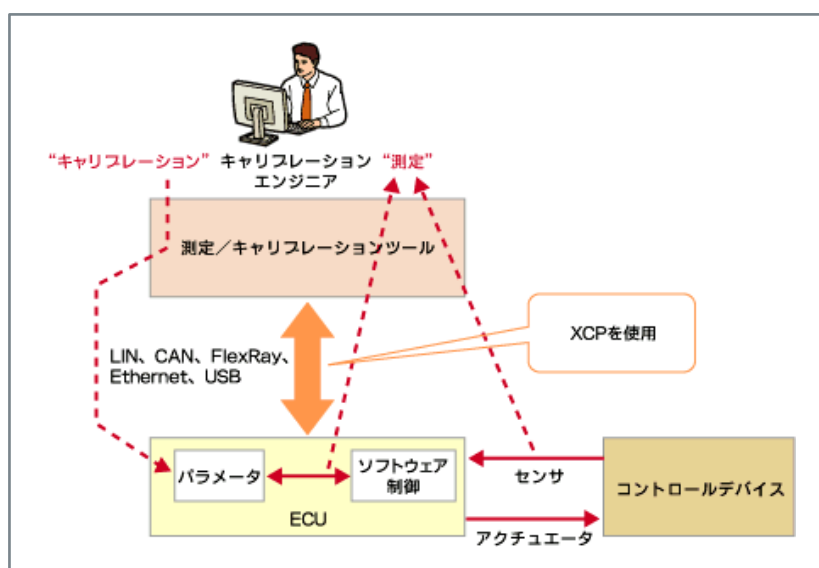


図 4: XCP

※ベクター・ジャパンの資料を基に作成

測定／キャリブレーションプロトコルの歴史

汎用的な測定／キャリブレーションプロトコルとして、最新のものが XCP であり、現在そのバージョンは 1.1 となっています。ここで、XCP 誕生の歴史的背景を簡単に紹介しておきます。

(1) CCP の誕生

キャリブレーションするパラメータおよび内部測定値へのアクセスを可能にするためのプロトコルとして、まず「CCP」プロトコルの最初のバージョン (Version 1.0) が、1992 年に規格化されました。CCP とは “CAN Calibration Protocol” の略であり、CAN を使った測定／キャリブレーションプロトコルという意味になります。CCP は、その後、機能が追加されてバージョンアップし、1999 年に Version 2.1 になりました。これが CCP の実質的な最終バージョンであり、それ以降の機能追加は XCP に引き継がれることになりました。

(2) CCP の普及

CCP は、CAN の普及とともに広く使用されるようになりました。その背景は、ECU 内部のソフトウェアへアクセスするための通信媒体として CAN を使うことで、低コストで汎用性のある測定／キャリブレーションが実現できたからです。実際、車載ネットワークに接続されている ECU に CCP ドライバを組み込むだけで、すでに設置されている CAN を使った ECU の測定／キャリブレーションが可能となります。

(3) CCP から XCP へ

CCP の普及後、ECU の性能の向上とともに測定／キャリブレーションに対する要求仕様が拡大していきました。例えば、CCP が規格化されたときに想定されていなかった測定速度や測定点数の拡大があります。さらに、CAN だけではなく、FlexRay など CAN 以外の車載ネットワークが規格化され、実際に搭載されるようになったこともその理由の 1 つです。

これらを踏まえ、CCP の仕様を引き継いだうえで新たな機能を追加し、さらには異なるネットワークでも対応できる汎用的な唯一の測定／キャリブレーションプロトコルとして、2003 年に XCP の Version 1.0 が規格化されました。

ASAM とは？

CCP および XCP は、「ASAM (Association for Standardisation of Automation and Measuring Systems)」という団体によって規格されています。この ASAM は、自動車および産業機器に関する開発プラットフォームの規格化を行っております。なお、規格化された内容は ASAM の Web サイトで確認できます。

CCP および XCP は、ASAM の持つ「AE (Automotive Electronics)」というワーキンググループの活動によって規格化されたもので「MCD」という領域に割り当てられています。この MCD という名称は “Measurement” “Calibration” “Diagnostic” の頭文字が由来です。

ASAM における CCP/XCP の位置付け

ASAM の標準規格書の Standard の AE カテゴリ中に「ASAM MCD-1 XCP」という規格書が存在しますが、これが XCP の規格書になります。CCP も同じように「ASAM MCD-1 CCP」として存在します。MCD の規格全体における CCP/XCP の位置付けを図 5 に示します。

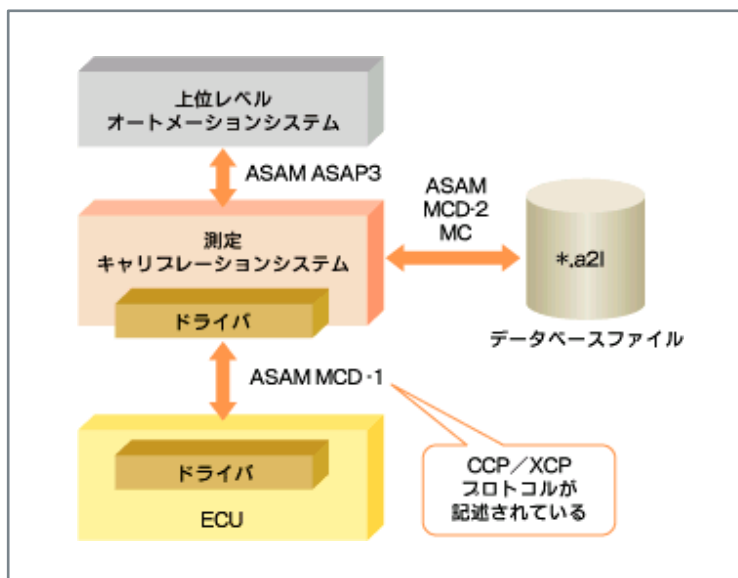


図 5: MCD 規格と CCP/XCP

※ベクター・ジャパンの資料を基に作成

CCP/XCP は、測定／キャリブレーションシステムであるツール側と ECU との通信プロトコルを定めたものになり、「ASAM MCD-2 MC」が以下 3 つの内容を持つデータベースファイルの規格になります。

1. CCP/XCP の通信設定情報

CAN を使う場合の CAN のボーレートやメッセージ ID など

2. 測定値情報

測定対象のアドレス、サイズや物理値変換係数など

3. パラメータ情報

キャリブレーションするパラメータのアドレス、サイズや物理変換係数など

つまり、CCP/XCP はプロトコルそのものを規定し、そのプロトコルの設定や測定／キャリブレーションする対象の情報は、別途データベースファイルがあり、ツールはデータベースファイルを基に ECU に対して、CCP/XCP を使ってアクセスを行います。

※補足: 図 5 にある「ASAM ASAP3」は、測定／キャリブレーションするシステムを外部からコントロールするためのインターフェイスとして規定されたものです。

XCP のメリット

続いて、XCP の持つ主なメリットについて紹介します。

(1) すべての ECU を統合的に測定／キャリブレーション可能

XCP は、おのものの ECU で使用するマイコンのバイトオーダー（エンディアン）や、使用できる ROM／RAM 容量の違いを吸収できるように考えられており、ネットワークとして使用する通信媒体が異なっても同一のプロトコルが使用できます。これにより、すべての ECU を XCP だけで測定／キャリブレーションすることが可能です。

(2) 実績のあるプロトコルを測定／キャリブレーションですぐに使える

XCP は、CCP の実績をベースに、ECU に対する測定／キャリブレーションへの要求仕様が盛り込まれています。また、プロトコルのすべてが ASAM の Web サイトで公開されており、誰でも使うことができます。また、ECU 側のプロトコル処理を行うソフトウェア・ドライバは、ツールメーカーによっては無償で提供されているケースもあります。

(3) ECU の動作に同期した測定が可能

一般的な ECU は、センサなどの外部情報や別の処理からの入力を基に処理を行い、別の処理や外部デバイスへ出力するという決められた制御周期で繰り返す、もしくは入力の要因となる事象をトリガにして処理を行うことになります。XCP により、この入力／出力のデータを、これら処理に同期して測定することが可能です（図 6）。

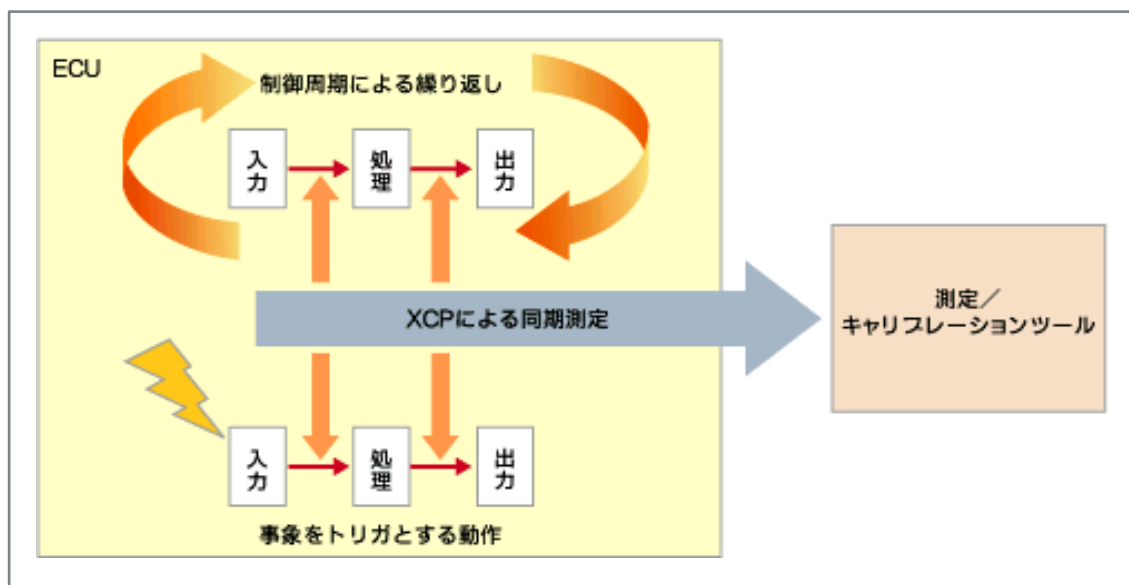


図 6: XCP による同期測定

※ベクター・ジャパンの資料を基に作成

測定／キャリブレーションを行うだけなら、車載ネットワーク上にあるほかのプロトコル、例えば車両診断に使われるプロトコルでも可能ですが、キャリブレーション工程において、制御に同期した測定は不可欠です。この制御に同期した測定も可能にするのが XCP というわけです。

XCP の各ネットワーク例

XCP は、ネットワークが異なっても同じプロトコルが使用されますが、実はネットワークごとにパケット構成などが規格化されています。これを「トランスポートレイヤ規格」といい、XCP + “on” + ネットワーク名

というルールで、各ネットワークで動く XCP に名前が付けられています。例えば、CAN で動く XCP は「XCP on CAN」となります。

最後に、XCP のネットワーク例を挙げます(表 2)。

XCPのネットワーク名	機械的キャリブレーション
XCP on CAN	CANを使ったXCPとなり、CCPの機能をすべて踏襲した新しいバージョンのCCPと見なすことができます
XCP on FlexRay	FlexRayを使ったXCPの最大の特徴としては、ネットワークにつながるECU間でLPDU (Logical Protocol Data Unit) を共用する仕様「Dynamic Band Width Management」が盛り込まれている点です
XCP on Ethernet	名称としてはEthernetとなりますが、Ethernet/パケットを使用するのではなくTCP/IP、またはUDP/IPを使用します。実際の運用としては、直接ECUにつなぐというケースよりもECUの専用インターフェイスをXCP on Ethernetに変換して使用するケースが多くなります

表 2: XCP のネットワーク例

さて、「測定／キャリブレーションとは何か？」からはじめ、プロトコル化の歴史やプロトコルの位置付け、メリット、ネットワーク例について解説しました。次は、実際の XCP のプロトコルの概要や代表的な機能について説明します。

2. XCP プロトコルの通信の仕組みと機能

さて、ここでは、「XCP(Universal Calibration Protocol)」プロトコルが具体的にどのように通信を行っているのかについて、XCP の持つ機能とプロトコル内容について説明していきます。

マスター・スレーブ方式

XCP では、測定／キャリブレーションを行うツール側を「XCP マスター」、測定される ECU 側を「XCP スレーブ」と呼び、いわゆる「マスター・スレーブ通信方式」が採用されています。

この通信方式は、マスターが必ず先にコマンド送信を開始し、マスターの送信をスレーブが受け取った後に、スレーブがマスターへレスポンスを送信するという通信順序となります。

図 7 に示すとおり、1 つのネットワーク上にマスターは必ず 1 つですが、スレーブは複数存在可能です。

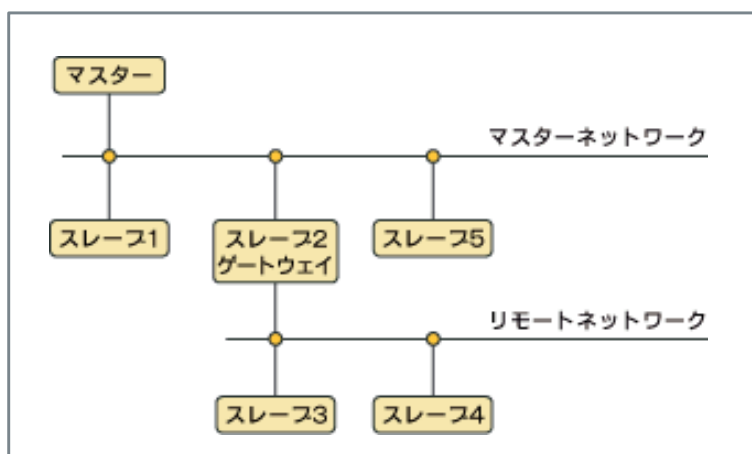


図 7: XCP を使用するネットワーク例

※ベクター・ジャパンの資料を基に作成

この場合、マスターは個々のスレーブに対してコマンドを送り、スレーブからのレスポンスを受け取ります(図 8)。

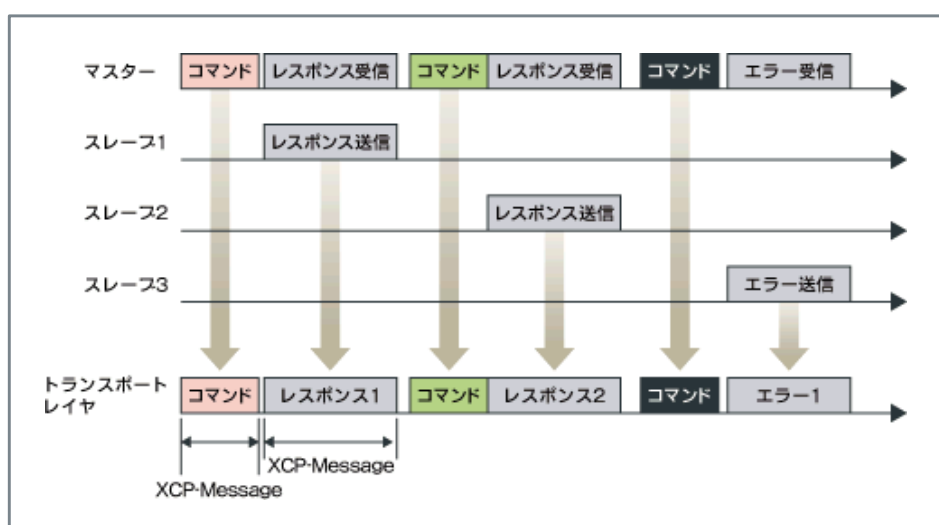


図 8: シングルマスター・マルチスレーブ通信例

※ベクター・ジャパンの資料を基に作成

この通信方式により、車載ネットワーク上に測定／キャリブレーションを行うツール(=XCP マスター)を 1 つ接続し、XCP を通じて測定対象となる各 ECU(=XCP スレーブ)にアクセスします。

ネットワークと転送方式

ネットワーク上において、“マスターからスレーブへの送信”と、“スレーブからマスターへの送信”の 2 種類を区別できる通信であれば XCP を使用できます。

「XCP on CAN」の場合、CAN の ID を 2 つ使い、“マスターからスレーブへ送信を行う ID”と“スレーブからマスターへ送信を行う ID”を割り当てることで区別します。

区別された通信をネットワーク上で使用し、何らかの内容を転送する際、XCP では 3 つの転送モードが用いられます(図 9)。

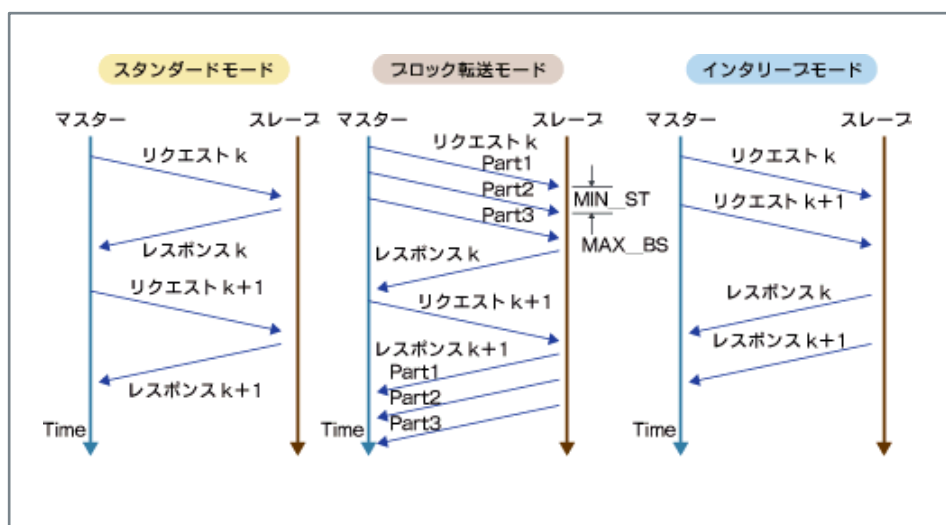


図 9: XCP の転送モード

※ベクター・ジャパンの資料を基に作成

この転送モードは、マスター側およびスレーブ側それぞれで、どのモードにするのかを決めることができます。例えば、マスターは「ブロック転送モード」、スレーブは「スタンダードモード」で、といった利用も可能です。これにより、マスターであるツール側は高機能だが、スレーブである ECU は限られたリソースしか使えない……といった際に、転送モードを単純なものにするといった実装も可能です。

CTO/DTO

XCP では、このマスターとスレーブ間の送信方向の違いとは別に、送信する内容を“XCP 自身の制御に関する通信”か、“データに関する通信”かの 2 種類に分け、それぞれのネットワーク上に流すパケットのフォーマットを定めています。ちなみに、前者を「CTO(Command Transfer Object)」、後者を「DTO(Data Transfer Object)」と呼びます。

● CTO: Command Transfer Object の略

CTO とは、XCP 自身の制御コマンドやそのレスポンスなどに関するオブジェクトです。制御コマンドはマスターから、コマンドに対するレスポンスはスレーブから送られます。

● DTO: Data Transfer Object の略

DTO とは、同期的なスレーブ (ECU) のデータ測定結果、および同期的なスレーブへのデータ変更に関するオブジェクトです。同期的なデータ変更を「ステミュレーション」と呼びますが、これは測定／キャリブレーション以外の機能になりますので詳細は割愛します(詳しく知りたい場合は、XCP の規格書をご覧ください)。

図 10 に、XCP の規格書 Part2 1.1.1 The XCP Packet Types から抜粋した XCP マスターとスレーブ間における CTO と DTO の関係を示します。

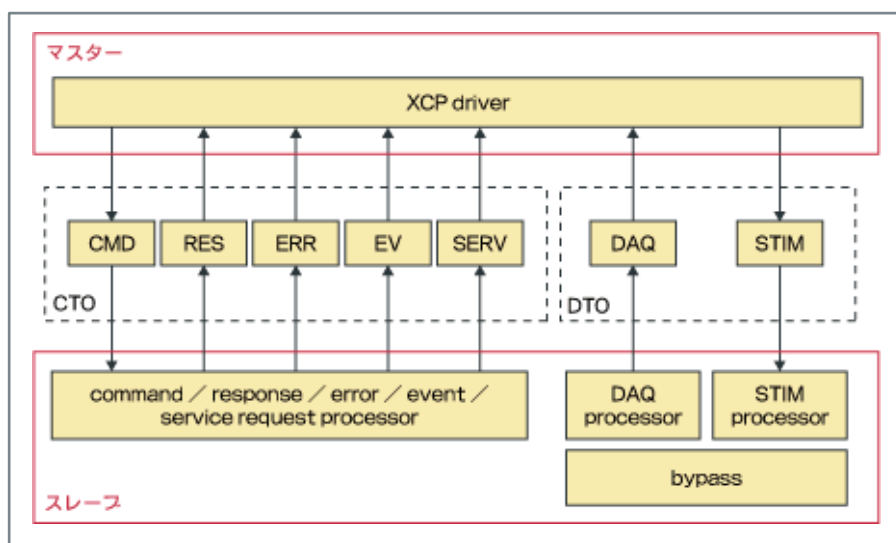


図 10: XCP マスター/スレーブ間オブジェクト

パケットフォーマットと PID

CTO と DTO は、前述のとおり異なるパケットフォーマットになり(図 11)、それぞれ最大パケット長をマスターとスレーブ間で設定できます。

- ・MAX_CTO: CTO の最大パケット長[バイト]
- ・MAX.DTO: DTO の最大パケット長[バイト]

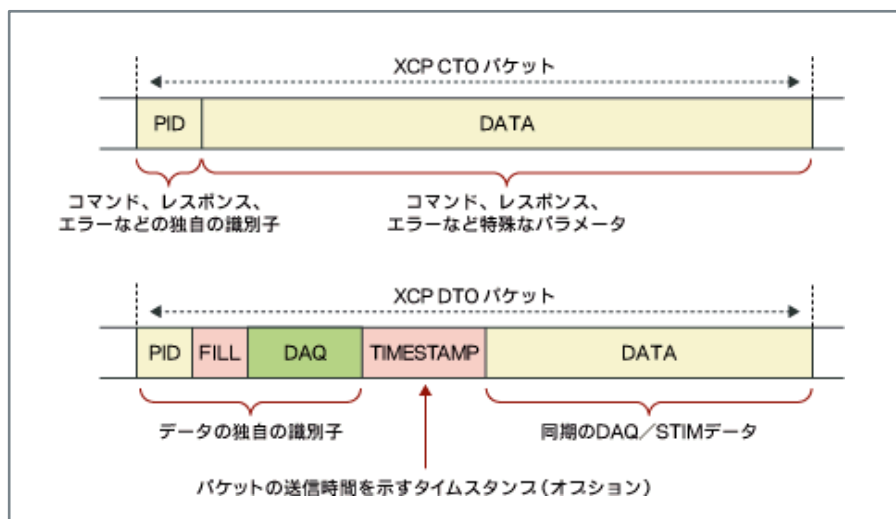


図 11: CTO と DTO パケットの構成

※ベクター・ジャパンの資料を基に作成

すべての XCP のパケットは、この最大パケット長以内に収まり、1 つのパケットでコマンドやレスポンスなどの内容が完結するようになっています。また、パケットフォーマット中、各フィールド 2 種類の送信を CTO、および DTO のパケットに割り当て、さらに、パケットの最初の 1 バイト目を「PID」と呼ばれる識別子とし、どのようなパケットであるかを区別するために使用しています。

マスターからスレーブへの XCP 制御を行う場合を「コマンド(CMD)」とし、その際 PID は「0xC0」～「0xFF」の範囲となります。このコマンドに、スレーブが肯定応答を返す場合を「レスポンス(RES)」とし、その際の PID は「0xFF」になります(図 12)。

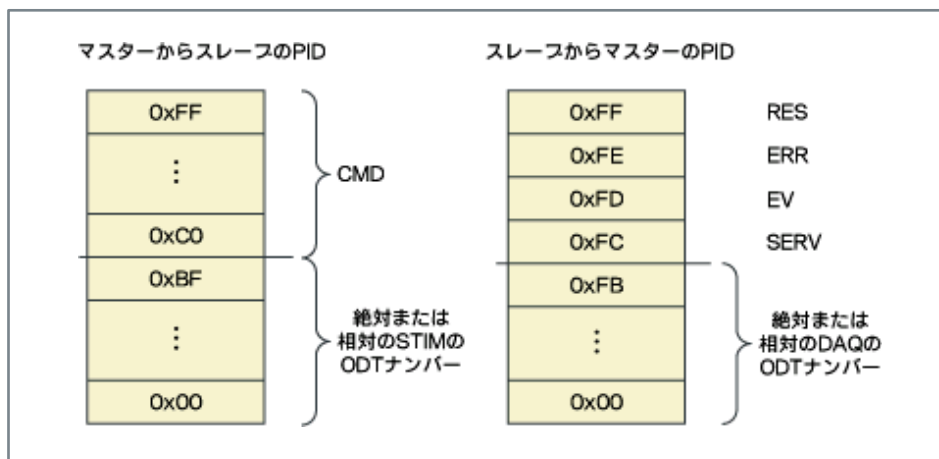


図 12: パケットの識別子(PID)

※ベクター・ジャパンの資料を基に作成

測定／キャリブレーションに関する同期的なデータのやりとり以外は、すべてマスターからのコマンドによって行われ、その肯定応答をスレーブが返すことになります。

ここで、マスターからスレーブの CAN ID を「1」、スレーブからマスターの CAN ID を「2」として XCP on CAN を使用した場合を例に挙げて説明します。

この場合、XCP の通信は以下のような順で行われます。

1. マスターが CAN ID の「1」で最初の 1 バイト目に「0xFF」を、2 バイト目に「コマンドパラメータ」を指定して送信
2. スレーブが 1 番を受信し、PID でコマンドを検知
3. スレーブが CAN ID の「2」で最初の 1 バイト目に「0xFF」、2 バイト目以降に「レスポンス値」を指定して送信
4. マスターが 3 番を受信

図 13 は、実際に通信が行われている様子をトレースした結果です。1 番の「0xFF」は、「CONNECT」というコマンドになり、これによってマスターとスレーブの論理的なコネクションを確立し、以降のコマンドを受け付けています。

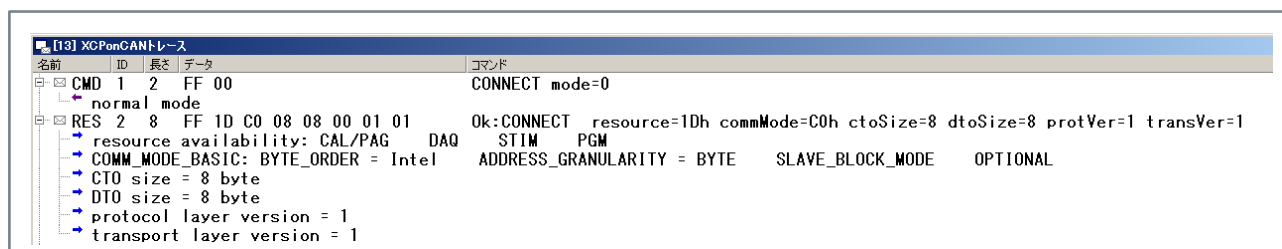


図 13: CONNECT コマンドとレスポンス

測定／キャリブレーション対象へのアクセス

XCP による測定／キャリブレーションは、ECU 内部のソフトウェアに対してアクセスすることで実現しています。具体的には、測定／キャリブレーション対象メモリ区間に対する「XCP アドレス」を指定する方式でアクセスが行われます。

XCP アドレスとは、通常のマイコンにおけるアドレスとほぼ同じですが、XCP では 32 ビットの XCP アドレスと 8 ビットのアドレス拡張の組み合わせによって指定します。つまり、マスターはスレーブに対して、 $32 + 8 = 40$ ビットアドレス空間へのアクセスが可能になります。

このアドレスは、XCP の実際の ECU メモリと必ずしも一致させる必要はなく、そのバイト順(エンディアン)は、スレーブごとに選択できます。そのため、車載ネットワーク上に複数の ECU がつながっており、それぞれ異なるアドレス空間(16 ビット、32 ビット)、異なるアドレスバイト順が存在したとしても、マスターがこれらを適切に扱うことで、すべてを測定／キャリブレーションできます。

非同期測定

XCP は、同期だけではなく非同期測定も可能です。

非同期測定は、マスターからのコマンドを使って、スレーブである ECU 内部のデータを XCP アドレス指定で取り出し、スレーブからの応答で、このデータをマスターに渡すという通信を繰り返すことによって実現します。取り出すためには、PID を「0xF4」とした「SHORT_UPLOAD」というコマンドを用います。このコマンドとレスポンスのフォーマットについて以下に説明します。

- SHORT_UPLOAD コマンド:
 - CTO 0 バイト位置 - PID として「0xF4」を指定
 - CTO 1 バイト位置 - 取り出すバイト数を指定。最大は MAX_CTO-1 バイト
 - CTO 2 バイト位置 - 予約領域
 - CTO 3 バイト位置 - 読み出すアドレス拡張 8 ビットを指定
 - CTO 4 バイト位置～7 バイト位置 - 読み出すアドレスを指定
- SHORT_UPLOAD レスポンス:
 - CTO 0 バイト位置 - PID として「0xFF」を指定
 - CTO 1 バイト位置～MAX_CTO バイト位置 - 取り出したデータを指定

図 14 は、マスターが「SHORT_UPLOAD」コマンドを使って、100ms ごとに XCP アドレス拡張を「0」、XCP アドレスを「0x00124A5C」から 4 バイト取り出している様子をトレースした結果です。

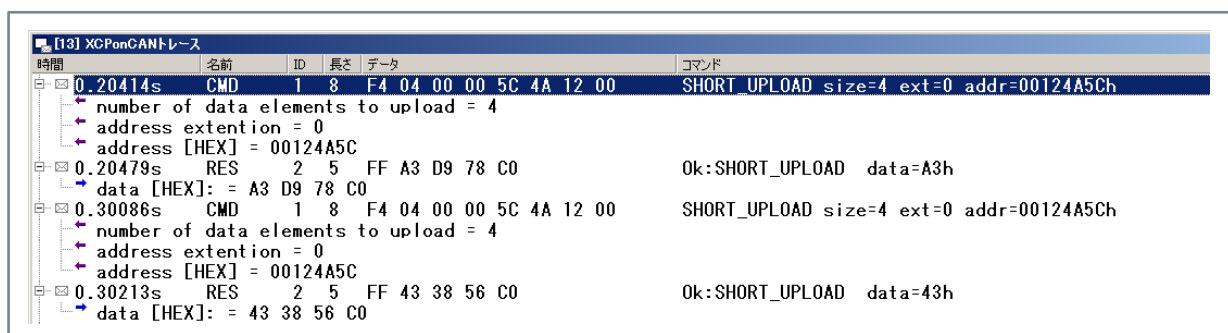


図 14: SHORT_UPLOAD コマンドによる測定

同期測定

前述の非同期測定の場合、測定タイミングはマスターが決定していました。ECU の制御に合わせた測定を可能にするには、測定タイミングを ECU が決定し、データを取り出した後に、スレーブからマスターへデータを渡すという処理が必要になります。このデータを渡すための通信は、DTO を介して行われます。

データを取り出す XCP アドレスは、マスターが同期測定の前にコマンドで指定し、同期測定の開始コマンドを待って、スレーブが DTO を使ってマスターに送信します。このため、同期測定の場合は、コマンドとレスポンスの組み合わせではなく、測定周期やイベントによって、スレーブから DTO パケットがマスターへ送信されます。図 15 に、「START_STOP_SYNC」コマンドで同期測定を開始し、スレーブが測定データを DTO パケットを使って送信して、「START_STOP_SYNC」コマンドで停止するまでを実際にトレースした結果を示します。

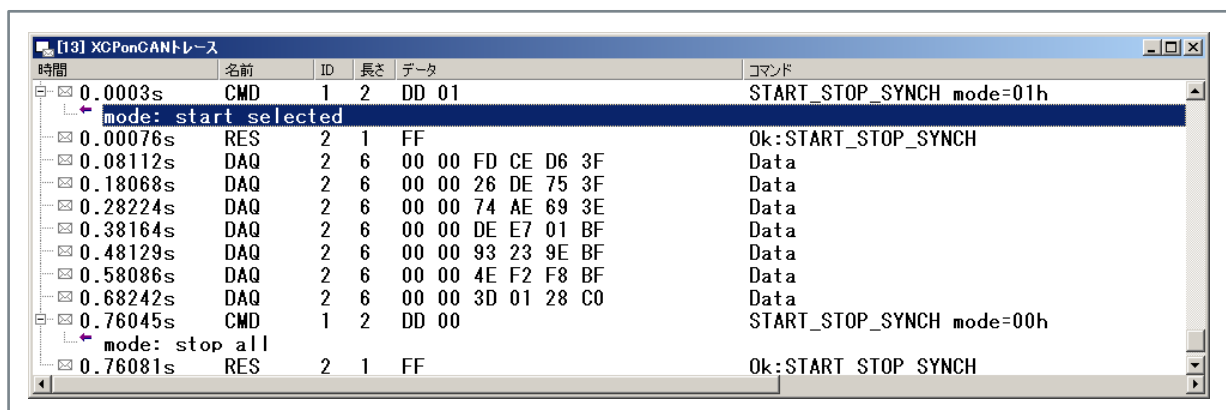


図 15: 同期測定の開始と停止

以下、同期測定における測定タイミングを決定する ECU の制御アプリケーション、マスターとスレーブの役割について説明します。

- ECU の制御アプリケーション:

測定したい制御周期、および事象のトリガが発生した際、それを XCP スレーブへ伝える処理を行います。

- マスター:

同期測定するメモリとその周期を決め、同期測定を行う XCP アドレスの指定、および同期測定の開始と停止をコマンドで行います。

- スレーブ:

マスターから指定された同期測定 XCP アドレスを管理します。また、同期測定の開始を検知したら、停止するまでは上記 ECU の制御アプリケーションからの通達ごとに管理した XCP アドレスのメモリ値を取り出して測定を行い、DTO パケットを送信します。

同期測定にかかわる XCP 用語と概念

同期測定に関して、XCP 規格書ではさまざまな用語が使われています。ここでは、主な用語とその概念について説明します。

- Element:
XCP アドレスによって指定される 1 つの測定対象メモリです。
- ODT (Object Description Table):
Element の測定したメモリを並べ、1 つの DTO パケットで最大送信できる分のメモリを集めたテーブルです。
- ODT エントリ:
ODT を作成するための Element の測定対象 XCP アドレスです。
- DAQ リスト:
ODT を集めたものです。1 つの同期測定の周期、または事象のトリガごとに測定されるメモリ量がこれで決まります。ODT は、1 つの DTO パケットなので、最大でも MAX.DTO-PID のサイズで決まりますが、実際に測定されるメモリは、これよりも大きくなることもあるので、ODT と DAQ リストに分けられています。
- イベントチャンネル:
同期測定を行う制御周期、および事象のトリガのチャンネル、つまり“種別”を意味します。同期測定するタイミングは、すべてこのイベントチャンネルを基にします。

同期測定時の処理

同期測定において、ECU の制御アプリケーションがスレーブに測定タイミングを伝えたとき、どのような処理が行われるのかを以下で説明します。

- (1) 制御アプリケーションが、スレーブに測定タイミングを通知します。この通知は、通常、スレーブのドライバ処理でイベントチャンネルを引数とする処理を持ち、これを呼び出すことで実現しています。
- (2) スレーブには、マスターから指定された測定対象メモリのアドレスを ODT エントリとして管理するバッファがあり、このバッファを 1 番の呼び出し時に確認し、指定アドレスのメモリを読み出します。
- (3) 読み出したメモリを ODT の単位で集めて、DAQ リストの分だけの DTO パケットを生成します。

図 16 が、Element と ODT エントリ、DAQ リストと ECU の制御アプリケーションの関係を表した例です。制御アプリケーションに Vector 社のスレーブドライバを組み込んだ場合、「XcpEvent(イベントチャンネル)」という C 言語関数を呼び出すことで、ODT エントリのバッファを基に、ECU メモリから Element が読み込まれ、DTO パケットが作られます。

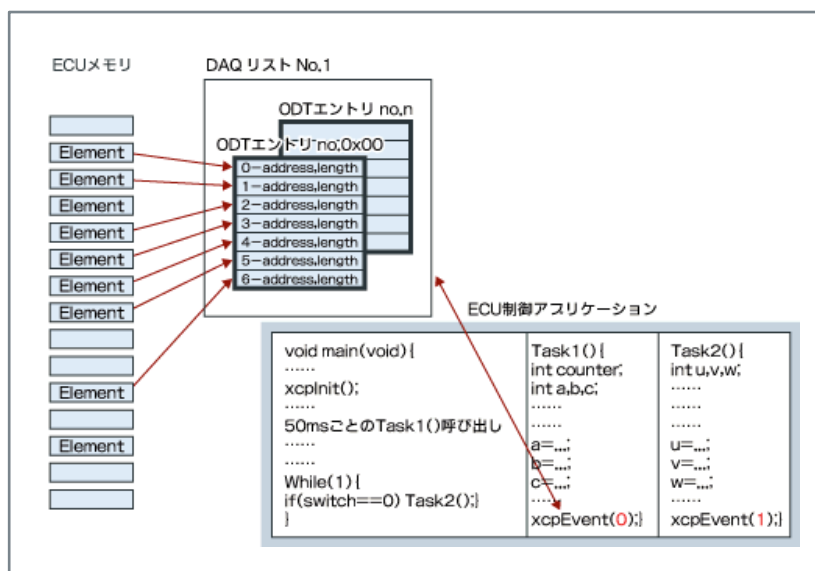


図 16: XCP と制御アプリケーションの関係

※ベクター・ジャパンの資料を基に作成

測定対象と DTO パケット

図 17 は、ECU のメモリ内の測定対象である Element が、DTO パケットとしてネットワーク上に送信されるまでの関係を示したものです。

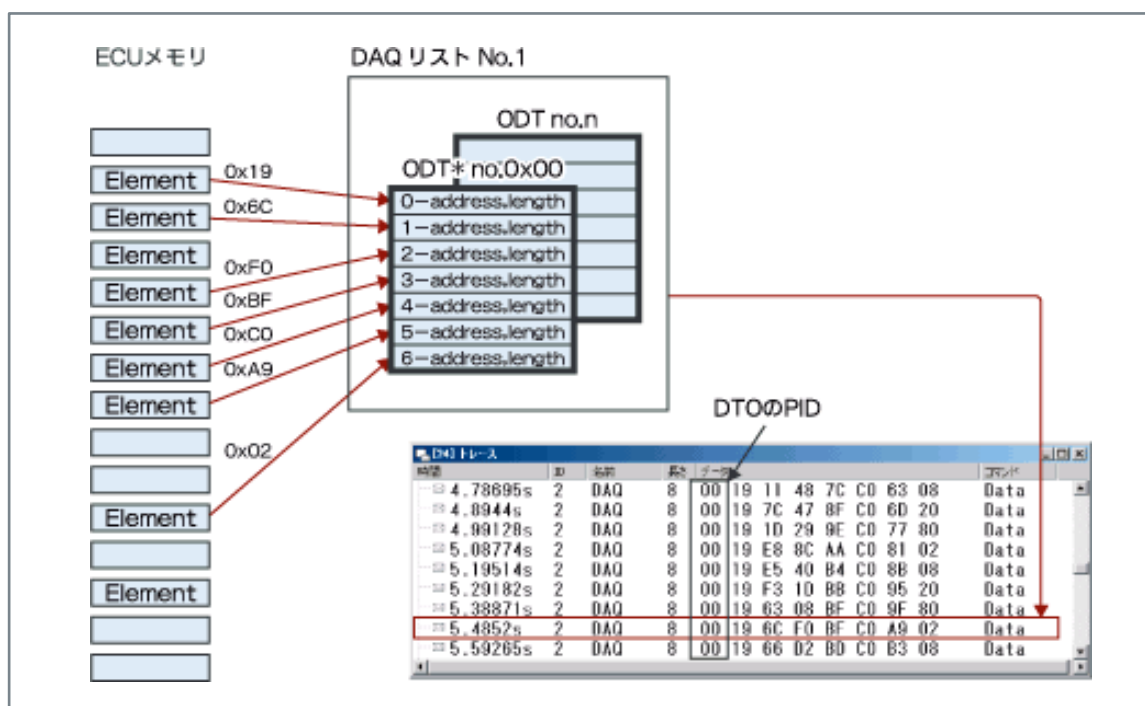


図 17: Element と DTO パケットの関係

※ベクター・ジャパンの資料を基に作成

図 17 では、Element を ODT エントリに従って 7 つ測定した結果、「0x19」「0x6C」「0xF0」「0xBF」「0xC0」「0xA9」「0x02」が取り出され、1 つの ODT として DTO パケットが作られています。そして、この DTO パケットには最初の 1 バイトに PID、2 バイト目以降には測定された時点での Element の内容が入っています。このように PID に「0」～「0xFB」の値を持つ DTO パケットを「DAQ」と呼びます。

このように、測定対象となる ECU メモリ、測定対象を管理する ODT から DAQ リスト、測定タイミングを決める制御アプリケーション、測定タイミングを区別するためのイベントチャンネルを基に、DTO パケットを使って測定データをマスターへ送ることで、同期測定を実現しています。

同期測定のオプション

XCP での同期測定には、これまで説明したもの以外にも測定規模・用途に応じて、さまざまな利用形態があります。これらの利用形態について、代表的なものを紹介します。

- ダイナミック DAQ:

測定対象を管理する DAQ リストを持つ ODT とそのエントリ数が多いほど、測定できる測定点数が増えることとなります。また、測定タイミングとなるイベントチャンネルの数だけ DAQ リストを持つことで、ECU のすべての測定タイミングに対する測定が可能になります。しかし、これらの数を増やすと ECU 内での管理バッファが増え、結果として ECU のメモリを消費することになります。また、ECU に 10[msec]と 25[msec]の制御周期があるような場合、ある測定では 10[msec]の制御周期での測定点数が多いが、別の測定では 25[msec]が多いなど、測定する場面によって測定点数が異なる場合があります。こうした用途向けに、スレーブ側で DAQ リスト、ODT、ODT エントリ数を測定ごとにダイナミックに変えられる機能「ダイナミック DAQ」と呼ばれるものがあります。それとは逆に、XCP ドライバの組み込み時に、あらかじめ数が固定化されているような場合は「スタティック DAQ」と呼びます。1 つのスレーブでスタティック／ダイナミック DAQ のどちらかの機能を持つこととなります。

- タイムスタンプ付き DAQ:

マスターとスレーブ間の通信において、ゲートウェイが入ることで時間差が生じるような場合、または無線通信などを使用しているため、通信時間に揺らぎが生じるような場合、同期測定では測定タイミングがマスターから見ると“不明”になります。これを防ぐために、「タイムスタンプ付き DAQ」が用いられます。スレーブ側は、測定時間のタイムスタンプ値を付けた DAQ を DTO パケットでマスターへ送信します。そして、これを受け取ったマスター側は、付加されたタイムスタンプ値を読み出すことで、正しい測定タイミングを知ることができます。

キャリブレーション

キャリブレーションは、ECU 内部ソフトウェアにあるパラメータを書き換えるため、マスターからの XCP アドレスの指定コマンド、および書き変えるデータを送信するコマンドを送り、スレーブがそれに応じて該当するパラメータのアドレスを導き出して、書き換えを実行し、レスポンスを返します。

図 18 は、XCP マスターによって XCP アドレス拡張を「0」、XCP アドレスを「0x0041D5C8」から 2 バイトのパラメータを「0x0064」で書き換えている様子をトレースした結果です。

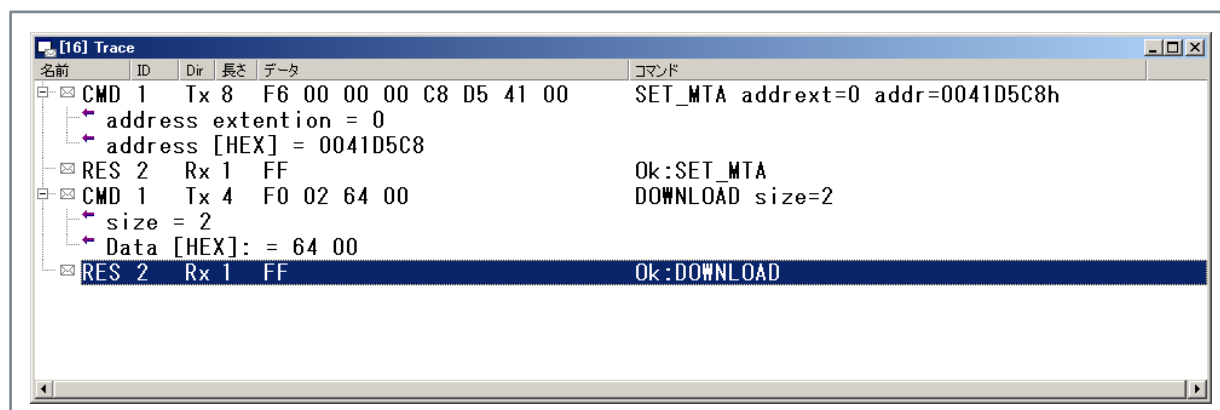


図 18: DOWNLOAD コマンドによるキャリブレーション

その他の XCP 機能

XCP には、測定／キャリブレーションだけでも数多くの機能が存在します。今回紹介したもの以外にも、ECU 内のフラッシュメモリのような不揮発メモリに対するアクセス機能や、ECU に対するアクセスをプロテクトする機能など、実に多くの機能が盛り込まれています。

以上、ここまでは XCP プロトコルの通信の仕組みや機能について詳しく説明しました。次は、「XCP の動向と適用事例」について紹介します。

3. XCP の動向と適用事例

では、XCP を使った測定／キャリブレーションのシステム構成からその応用例、動向と適用事例について説明します。

測定／キャリブレーションのシステム構成

これまでの説明にあるとおり、XCP はマスター側である測定／キャリブレーションツールと、スレーブ側である ECU の通信プロトコルを定めたものになります。

XCP のマスター側は、一般的に PC 側で動作するアプリケーションとして提供されます(図 19)。それに対して、スレーブ側は ECU に組み込まれる C 言語のソースコードの形でモジュール提供され、実際のトランスポート層、例えば XCP on CAN の場合、CAN の通信部分とつながって XCP のプロトコル処理を行うように実装されます。

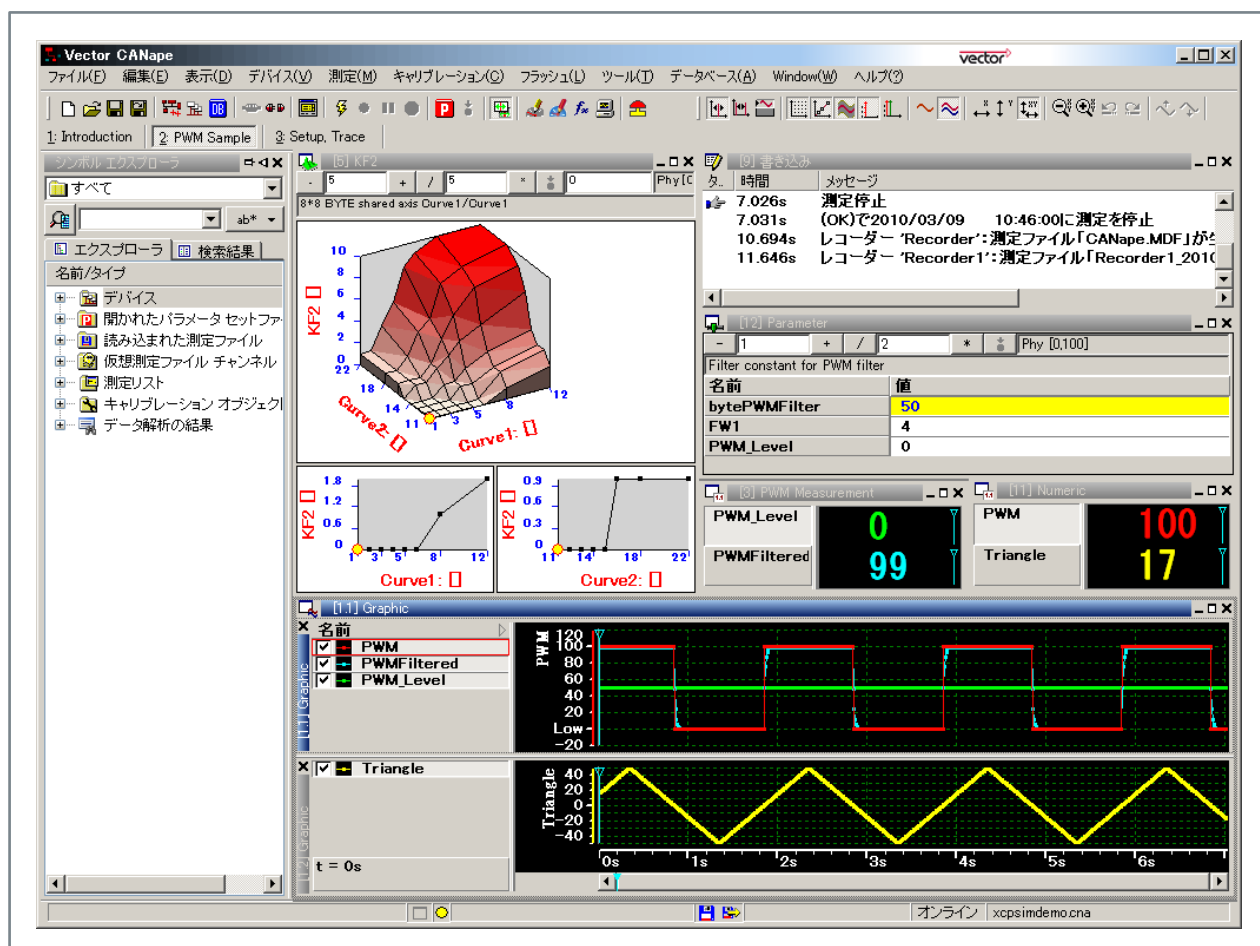


図 19: 測定／キャリブレーションツール

※画像提供: ベクター・ジャパン

XCP スレーブドライバとその提供

XCP のスレーブ側のモジュールは、一般的に XCP スレーブドライバとして XCP のプロトコル処理部分、および同期測定の実行部分を持ちますが、このドライバは通常 XCP のマスター側のアプリケーションを提供するツールベンダが提供しています。従って、測定／キャリブレーションツールベンダが、スレーブドライバの提供や実際に組み込むサービスなどを行うことになります。

総合化された測定／キャリブレーションシステム

キャリブレーションの目的である制御全体の最適化を行って要求定義に合わせるという工程は、XCP による ECU の測定／キャリブレーションだけでは実現できません。例えば、本稿の「1. 測定／キャリブレーションプロトコルとは？」のキャブレターのキャリブレーションには、混合気の状態をセンサなどで測定する必要があります。このようなセンサ入力を実際の測定／キャリブレーションの実現には必要不可欠になります。

さらに、近年では自動車の制御も高度化・高機能化が進み、車両の外部環境を基に制御を行うような ECU が出てきました。「ACC」と呼ばれる「アダプティブ・クルーズ・コントロール」がその代表的な例です。この ECU では、電磁波やカメラを使った車両の外部環境の検出による車両制御、例えばミリ波レーダーによる先行車両の検出を行って、その追従走行や衝突回避などを行っています。この場合ですと、外部環境である車両の位置やそのときの状況を映像などで測定する必要があります(図 20)。

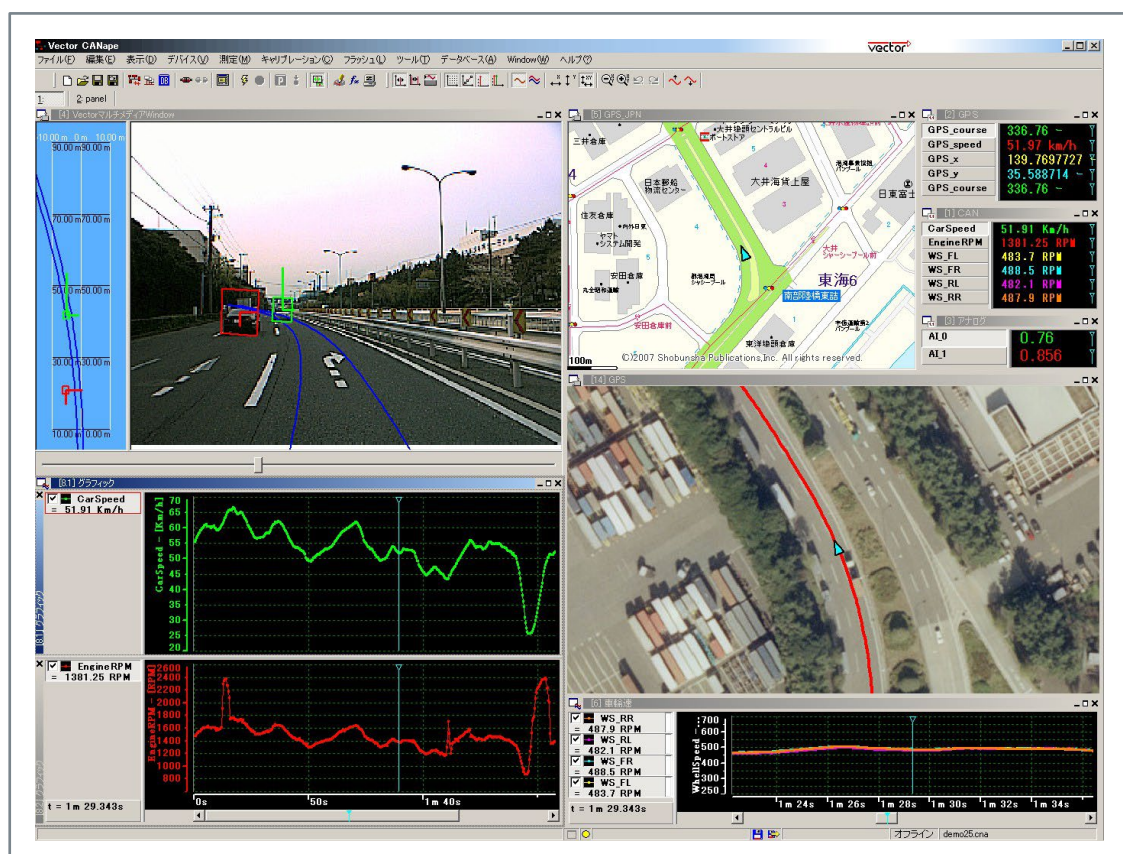


図 20:統合化された測定／キャリブレーションツール

※画像提供:ベクター・ジャパン

このようにキャリブレーションの実現には、統合化された測定／キャリブレーションシステムが必要になってきます。続いて、キャリブレーション工程において副次的に求められる主な要求項目について説明します。

- 統合的な測定

ECU の測定だけでなく、センサや車載ネットワーク、アナログ／デジタル信号や ECU によっては車両の位置や周辺の映像を同期して測定することが求められます。

- 測定のオフライン評価

キャリブレーションした結果や制御の妥当性を確認するために測定が行われますが、XCP にかかわる自動車および一般産業機器などでは測定に掛かるコストが高くなるケースが見られます。例えば、テストカーを使用したテストドライバによる測定はそれだけのコストが掛かりますし、航空機のエンジン測定などはさらに多くのコストが掛かることが容易に想像できます。このために、測定機能の高度化とともに、測定後のデータ解析や測定とキャリブレーションの因果関係の記録など、オフラインつまり測定後の評価機能が求められます。

- キャリブレーションしたパラメータの管理

制御にかかわる機能がすべて網羅されている ECU であれば、制御対象が変わったとしてもキャリブレーションによってパラメータを変えるだけで、制御を変えることなく使用できます。エンジン制御では、部品という形でメーカーから販売され、購入後にパラメータを実際のエンジンに合わせてキャリブレーションすることで使用できる ECU も存在します。

このような ECU をさまざまな制御対象に合わせてパラメータの変更で共用化するような場合、制御対象ごとの適切なパラメータの集まり(パラメータセット)を管理する必要が出てきます。

さらにこのような場合、より多くの制御対象に合わせるためにパラメータの数は多くなり、結果としてキャリブレーションの複雑化や複数のキャリブレーションエンジニアによる作業なども発生します。そのため、こうした側面からもパラメータの管理は必要になります。

パラメータの管理としては、多くのパラメータセットの比較やマージ、検索などの機能が求められます(図 21)。

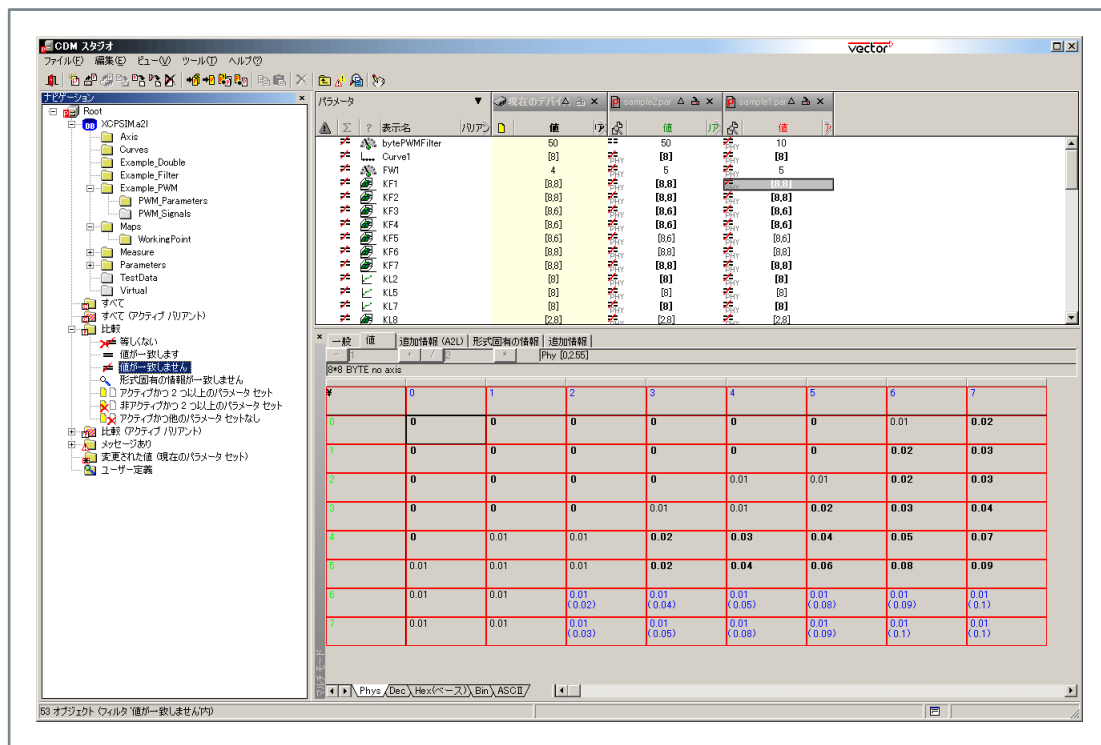


図 21: キャリブレーションしたパラメータの管理例

※画像提供: ベクター・ジャパン

XCP の応用事例

XCP は、これまでどおり測定／キャリブレーションシステムと、対象となる ECU およびそのネットワークにおいて使用されていますが、これ以外での XCP の利用について説明します。

(1) ロガーによる XCP ロギング

車載ネットワークや車両のアナログ状況を記録するロガーにおいて、これと合わせて「ECU の内部状態もロギングしたい」というニーズに応えるために XCP 対応している製品が存在します。アナログや車載ネットワークという出力結果のみならず、ECU の内部状態という出力過程もロギングすることが可能になり、解析効率が上がります。

(2) テスト

ECU の外部環境を電氣的に模擬し、ECU の動作テストを行うような場合において、そのテスト結果の判定や解析要因として、XCPを使用した ECU の内部状態の測定を行うシステムが存在します(図 22)。これにより、最終的な ECU の出力結果だけでなく、過程も確認することが可能になり、テスト時の解析効率が上がります。

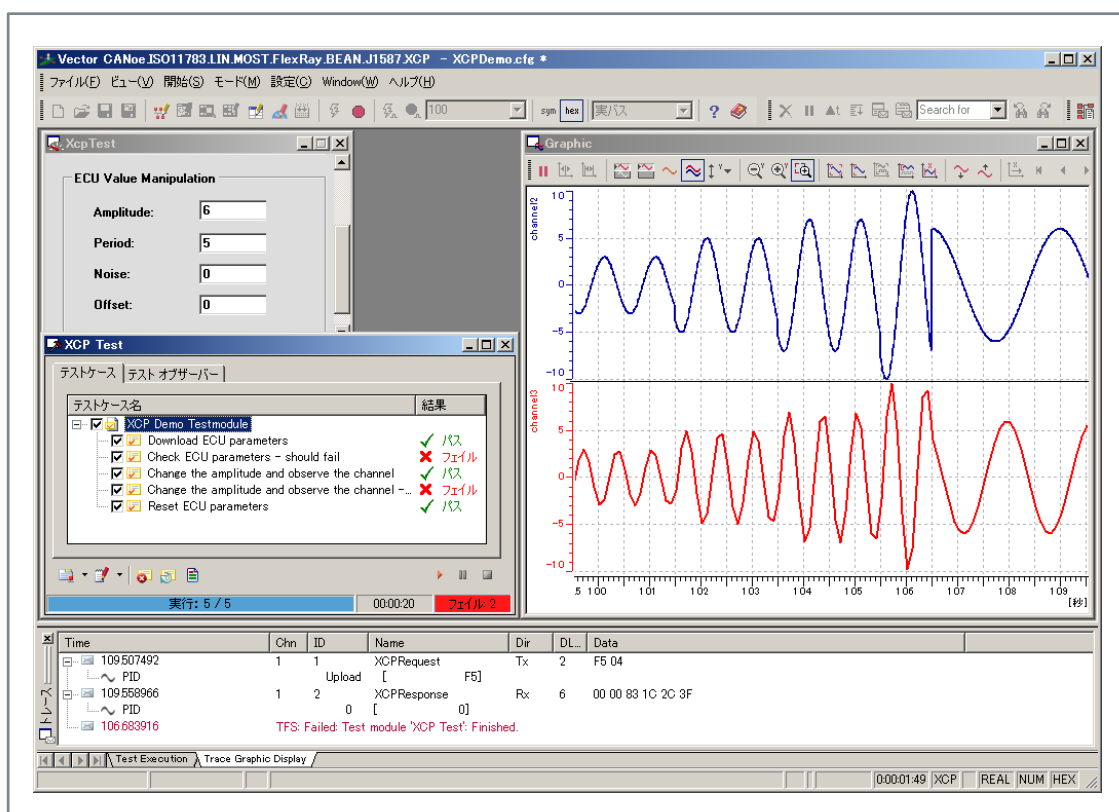


図 22: ベクターの CANoe による XCP を使った ECU のテスト実行

※画像提供: ベクター・ジャパン

(3) 測定器の汎用インターフェイス

本稿の「2. XCP プロトコルの通信の仕組みと機能」の説明のとおり、XCP ではマスターからの 40 ビットの XCP アドレスによる測定／キャリブレーション対象へのアクセスとその応答、そして同期測定対象を XCP アドレスによる指定とスレーブからの周期的な測定結果データの送信によって実現しています。この XCP を測定器の汎用インターフェイスとして使用している製品が存在します。この場合、XCP アドレスは ECU のメモリを指すのではなく、測定対象のアナログやセンサの識別用番号として使われます。

このように、XCP を測定器の汎用インターフェイスとして使用することで、既存の測定／キャリブレーションシステムに何ら変更を加えることなく、総合的な測定対象の 1 デバイスとして使用することが可能になります。

EV/HEV 自動車開発における XCP の利用

次に、EV/HEV 自動車の開発において考えられる XCP を使った測定/キャリブレーション要件について説明します。

(1) 複数 ECU の測定/キャリブレーション

HEV の場合、動力源がエンジンとモータの 2 つに増え、EV/HEV ともにバッテリーや回生充電など統合的な制御が求められます。これはシングルマスター、マルチスレーブの XCP の機能に加えて、ネットワークが分離された状態においてネットワークごとにマスターになれる測定/キャリブレーションツールが求められます。

(2) 高速測定を低負荷かつ安定した負荷で

モータの制御周期は 100[μsec]程度が要求され、これに同期した高速な測定が求められます。さらに、同期測定そのものの負荷、つまり XCP の測定処理負荷もこの場合できる限り低いことが求められます。通常ですと測定処理負荷は、測定点数に比例した一定量が測定周期ごとに掛かることとなりますから、測定周期がこのように高速になるほど、負荷が本来の制御周期に及ぼす影響は大きくなりますが、モータなどでは制御の周期がこの影響で揺らぐようなことは許されません。つまり、低負荷であると同時に測定点数などの測定状況の変化において、安定した負荷であることが求められます。

適用事例:

XCP on Ethernet とベクターの VX1000 システムを使った測定

前述の高速測定を行いつつ、低負荷かつ安定した負荷での測定を実装する事例としてベクターの「VX1000 システム」について説明します(図 23)。

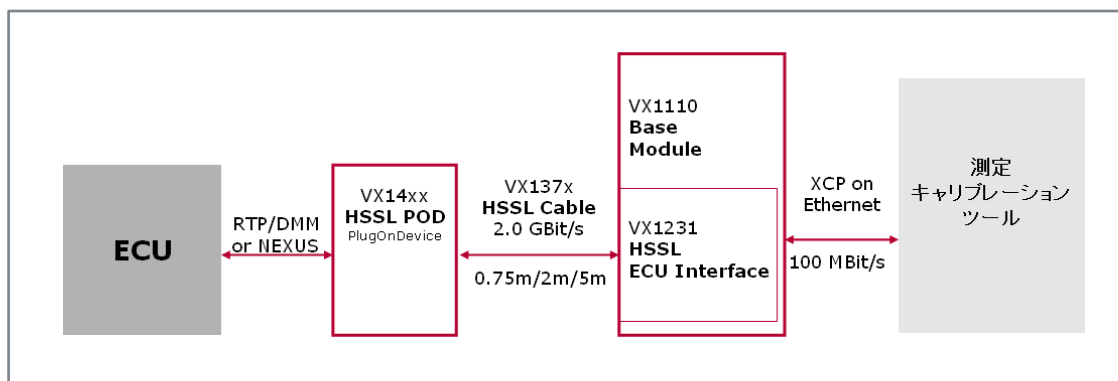


図 23: VX1000 システムの構成例

※画像提供: ベクター・ジャパン

このシステムでは、ECU に直接 XCP スレーブドライバを組み込まず、同期測定のタイミングを示す処理を行うドライバだけを使用します。このドライバを使って、ECU と VX1000 システムとは Nexus Class3 や RTP、DMM と呼ばれるマイコン内部のメモリトレースとして使えるインターフェイスでつながり、VX1000 システムと測定/キャリブレーションツールは XCP on Ethernet(TCP/IP または UDP/IP を使用)でつながります。測定/キャリブレーションツールからは XCP に見えますが、ECU では直接 XCP スレーブの処理を行わずにプロトコルの変換処理を VX1000 システムが行います。

続いて、ECU の測定周期のタイミングにおいて、ドライバの処理からどのように XCP の同期測定となるかについて説明します。

1. XCP と同様に制御アプリケーションが測定のタイミングを通知します。通知の方法についても、ドライバの関数呼び出しとイベントチャンネルを引数として持つ点も同じです。
2. 1 番の関数中では測定点ごとの処理を一切行わず、単純にタイミングのみを Nexus Class3 インターフェイスで通知します。このため、この処理時間は測定点数に影響されないうえに、低負荷になります。
3. VX1000 システム側で 2 番のタイミングを検出し、その時点でのトレースデータを基に XCP の DTO パケットを作り、XCP on Ethernet を使用してマスターに送信します。

各 XCP ネットワークプロトコルを直接使った場合と VX1000 システムの違いについて表 3 に示します。

ECUインターフェイス	ECU追加ソフトウェア	測定速度	ECUの負荷
XCP on CAN	XCPスレーブドライバ	50 [KBPS]	中程度
XCP on FlexRay	XCPスレーブドライバ	50~400 [KBPS]	高負荷
VX1000システム	測定タイミング処理のみ	5000 [KBPS]	非常に低負荷

表 3: ECU インターフェイスごとの特徴

モデルベース開発における XCP の利用

モデルベース開発(MBD)においては、実行可能なモデルを「プラント」と呼ばれる制御対象とともにシミュレーションすることができます。このシミュレーション実行の精度が上がってくると、実物と同じようにキャリブレーションを行うことも可能になります。さらに、キャリブレーションを行ってプラントに合致したモデルにすることで十分なシミュレーションができるようになる必要性もあります。このようにシミュレーション段階での測定とキャリブレーションが可能、かつ必要というのがモデルベース開発における要件の 1 つになります。モデルはシミュレーションだけでなく、自動コード生成によって C 言語などのコードに変換され、実際に動作することになりますが、この場合において従来と同じように測定／キャリブレーションが必要になり、さらには前述のシミュレーション段階でのキャリブレーション結果を基にしたキャリブレーションができること、つまり、キャリブレーションパラメータの引き継ぎができることが望ましくなります。

以下でシミュレーションと実物という 2 つの局面での測定／キャリブレーションの実現について考えてみます。

● シミュレーション

測定／キャリブレーション対象は、シミュレーションされた仮想モデルであり、これに対する仮想インターフェイスによって実現する必要があります。

● 実物

測定／キャリブレーション対象は、ECU 化されており、これに対する CAN や FlexRay などの実ネットワークを使って実現する必要があります。このようなシミュレーションと実物の違いについても XCP はネットワークに依存しませんから、このどちらでも測定／キャリブレーションが可能になり、モデルベース開発のすべての段階での利用が可能です。

適用事例:

ベクターの「CANape」と「Simulink XCP Server」を使った測定／キャリブレーション

シミュレーション時におけるモデルへの測定／キャリブレーションを行う事例として、ベクターの「Simulink XCP Server」について説明します(図 24)。

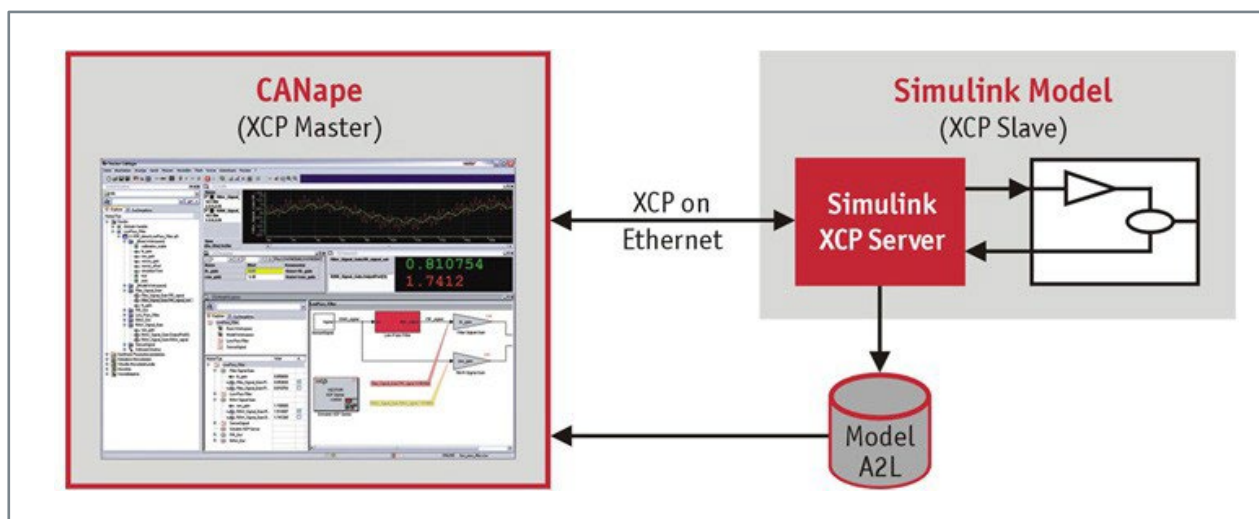


図 24: CANape と Simulink XCP Server

※画像提供: ベクター・ジャパン

Simulink XCP Server は、MATLAB/Simulink で動作しているシミュレーションモデルの測定／キャリブレーションが実現できるように、XCP のインターフェイスをモデルに設けるための Simulink のブロックであり、ベクターのキャリブレーションツール「CANape」と XCP on Ethernet を通じてつながります。これにより、実物の環境とまったく同じ測定／キャリブレーションをシミュレーション時でも利用できます。一般的に、シミュレーションは実物の実行時間よりも速く動作しますから、実行時間よりも速くキャリブレーションを行うことができ、このときに求められたキャリブレーションパラメータはシミュレーションの精度にも依存しますが、実物とほぼ同じ値であり、これを基に実物での測定からはじめることで、最終的なキャリブレーション完了までの時間を早めることができます。

XCP は、ネットワークに依存せず、どのような形であっても ECU へのアクセスが可能な状態からの測定／キャリブレーションを実現するために開発・規格化されました。

元来は、CCP の成功実績からその機能を拡張し、あらゆる車載ネットワークにおいても測定／キャリブレーションを実現することが主目的でしたが、その柔軟性から測定器の汎用インターフェイスとして使用され、さらにはモデルベース開発におけるシミュレーション時と実物時の違いにおいても同じインターフェイスとして利用されるようになっています。

XCP は、今後も測定／キャリブレーションのあらゆる局面において利用されることが予想されます。本稿が、皆さまの XCP についての理解と導入の一助となれば幸いです。

4. お問い合わせ窓口

技術関連のお問い合わせは、ベクターサポートまでお寄せください。

- ✓ サポートリクエスト(Web サイト経由のサポート問い合わせフォーム)
www.vector.com/jp/ja/support-downloads/support/

【お問い合わせ時のお願い】

お問い合わせの際、サポート開始前に製品のバージョン(例:CANape バージョン xx.x など)、シリアルナンバーなどをお伺いいたします。お手数ではございますが、事前にご確認のうえ、お問い合わせくださいますようお願い申し上げます。

【サポート対応について】

夜間／休日に頂いたお問い合わせは翌営業日の対応となります。また、内容によりご回答までにお時間を頂く場合がございます。なお、不具合が発生した際にご使用されていた関連プログラム一式のコピーを、検証のためご提供いただくようお願いする場合がございます。何卒ご理解ご協力のほど、よろしくお願い申し上げます。

- ✓ お見積もり／保守契約関連のお問い合わせ

営業部 TEL:(東京) 03-4586-1808 E-mail:sales@jp.vector.com



MORE INFORMATION

Visit our website for:

- > News
- > Products
- > Demo software
- > Support
- > Training and workshops
- > Contact addresses

vector.com