

次世代自動車基礎教材

はじめに

「太陽光発電--->環境エネルギー」「燃料電池--->クリーンエネルギー」それらを利用するスマートハウス・スマートシティ・スマートグリッド。それらと関係の深いデジタル家電は、スマートハウスに組み込まれて、情報家電としても本領を発揮することとなってきています。

一方、化石燃料を多く消費してきた自動車も大きな進化を遂げてHV（ハイブリッド自動車）・PHV（プラグインハイブリッド自動車）・EV（電気自動車）が実用化され、2014年12月には、ついにFCV（燃料電池車）が市販されることとなりました。このような自動車は次世代自動車として位置づけられ、スマートハウスやスマートシティで電気エネルギーを蓄積あるいは、発電する機器としても利用されるようになってきています。

スマートハウスや、次世代自動車、スマートシティやスマートグリッドというのは、デジタル制御を抜きには実現できず、デジタル家電や情報家電、自動車工学や組込系システム開発を学ぼうとする学生に、それぞれの橋渡しをするテキストという位置づけでこの本を作成しました。

本書は、どの章もほぼ独立しているので章順に読み進める必要はありません。気になる部分から読み始めてください。制御系・組込系システム開発を学ぶ方には、8章3節の「GPSセンサーの簡単な実験」と9章3節の「デジタルメータの開発」をお試し下さい。また、自動車工学や組込・制御系のシステム開発を学ぶ方には、5～7章の自動車に用いられる通信や自己診断システム、センサーや制御に関する記述が役立つと思います。

東日本大震災の復興を担う人材育成の中で、本書が少しでも役立てば幸いです。

プロジェクト開発分科会

開発担当：有限会社ワイズマン 原田賢一

目次

1章	スマートハウス	9
1章1節	「スマートハウス」とは？	9
1章2節	創エネ機器	11
1. 2. 1.	電池	11
1. 2. 2.	太陽電池	12
1. 2. 3.	燃料電池	12
1. 2. 4.	ガス発電・給湯暖房システム	12
1章3節	蓄エネ機器	13
1章4節	省エネ機器	14
1. 4. 1.	トップランナー制度	14
1. 4. 2.	省エネラベリング制度	15
1. 4. 3.	統一省エネルギーラベル	15
1章5節	HEMS	16
1章6節	スマートメーター	17
1章7節	スマートグリッド	19
1章8節	スマートシティ	20
2章	太陽光発電システム	23
2章1節	地球上の環境エネルギー	23
2章2節	太陽光発電の原理	24
2章3節	太陽電池の材料	26
2章4節	太陽電池の構成	27
2章5節	太陽電池の特性	29
2. 5. 1.	出力特性	29
2. 5. 2.	最大出力点	29
2. 5. 3.	温度特性	30
2. 5. 4.	放射照度特性	30
2. 5. 5.	変換効率	31
2章6節	太陽光電池の効率的運用	33
2章7節	住宅用太陽光発電システム	37
2. 7. 1.	規模による分類	38
3章	燃料電池	39
3章1節	家庭用燃料電池の概要	39
3. 1. 1.	ロゴマーク	39
3. 1. 2.	エネファームの仕組み	39

3. 1. 3. エネファームのメリット	40
3. 1. 4. エネファームのデメリット	41
3章2節 家庭用燃料電池の種類	42
3. 2. 1. 固体高分子形燃料電池 (PEFC : Polymer Electrolyte Fuel Cell)	42
3. 2. 2. リン酸形燃料電池 (PAFC : Phosphoric Acid Cell)	44
3. 2. 3. 熔融炭酸塩形燃料電池 (MCFC : Molten Carbonate Fuel Cell)	46
3. 2. 4. 固体酸化物形燃料電池 (SOFC : Solid Oxide Fuel Cell)	48
3章3節 燃料電池車 (FCV)	50
3. 3. 1. 燃料電池車のしくみ	50
3. 3. 2. 燃料電池車・電気自動車・ハイブリッドカーの違い	50
3. 3. 3. 燃料電池車 (Fuel Cell Vehicle)	51
3. 3. 4. 電気自動車 (Electric Vehicle)	51
3. 3. 5. ハイブリッドカー (Hybrid Vehicle)	51
4章 蓄電池	53
4章1節 2つの蓄電池	53
4. 1. 1. 鉛蓄電池	53
4. 1. 2. リチウムイオン蓄電池	58
4章2節 V2H (Vehicle to Home)	63
4. 2. 1. スマートハウスでのV2H	64
4. 2. 2. 自動車メーカーの取組	65
5章 自己診断システム	67
5章1節 自動車の自己診断機能	67
5章2節 ECU との通信規格	68
5. 2. 1. OBD1	68
5. 2. 2. OBD2	68
5章3節 OBD2の故障コード体系	70
5章4節 フリーズ・フレーム・データ	73
5章5節 OBD スキャンツール	74
5. 5. 1. スキャンツールの種類	74
5. 5. 2. DTC の読み取りと消去	75
5. 5. 3. ライブデータ表示機能	76
6章 自動車と通信	77
6章1節 CAN	77
6. 1. 1. CAN の歴史	77
6. 1. 2. ECU 間通信が求められた理由	77
6. 1. 3. 配線方式の違いによるメリット	78

6. 1. 4.	各ネットワークの位置付け	81
6. 1. 5.	ライン型構造	82
6. 1. 6.	マルチマスター方式	83
6. 1. 7.	CSMA/CA	83
6. 1. 8.	IDを使用したメッセージ・アドレッシング	84
6. 1. 9.	耐ノイズ性に優れた物理層	85
6. 1. 10.	エラー検出メカニズム	85
6. 1. 11.	データの一貫性	86
6. 1. 12.	通信速度など	86
6. 1. 13.	CAN の基本	86
6. 1. 14.	データフレーム標準フォーマット	89
6. 1. 15.	拡張フォーマット	92
6. 1. 16.	リモートフレーム	93
6. 1. 17.	データフレームとリモートフレーム	93
6. 1. 18.	通信調停	95
6章2節	LIN (Local Interconnect Network)	97
6. 2. 1.	LIN 策定の背景	97
6. 2. 2.	LIN の特徴と適応分野	98
6. 2. 3.	LIN ハードウェア	99
6. 2. 4.	通信方式	101
6. 2. 5.	ライン型バス構造	102
6. 2. 6.	マスター・スレーブ方式	102
6. 2. 7.	スケジューリング	103
6. 2. 8.	同期方法	104
6. 2. 9.	フレーム構造	105
6. 2. 10.	時間規定	108
6. 2. 11.	フレームタイプ	110
6. 2. 12.	エラー処理	112
6章3節	FlexRay	113
6. 3. 1.	なぜ新しい通信プロトコルが求められているのか?	113
6. 3. 2.	これまでの経緯	116
6. 3. 3.	FlexRay コンソーシアム	117
6. 3. 4.	FlexRay プロトコルの特徴	117
6. 3. 5.	通信構造／バスアクセス	123
6. 3. 6.	タイミング	128
6. 3. 7.	フレームフォーマット	129

6章4節	シリアル通信 (RS-232c)	133
6. 4. 1.	シリアル通信とは	133
6. 4. 2.	RS-232c とは	134
6. 4. 3.	RS-232c の信号レベル	135
6. 4. 4.	シリアル通信のイメージ	135
6. 4. 5.	信号の種類とコネクタ	136
6. 4. 6.	通信手順と実際の接続	138
6. 4. 7.	信号線の中のデータ	140
6. 4. 8.	シリアル通信の実験	141
7章	自動車と制御	143
7章1節	電気自動車の制御技術体系	143
7章2節	ハイブリッド電気自動車特有の制御	145
7章3節	システム全体の制御	146
7. 3. 1.	システムマネジメント制御	146
7. 3. 2.	協調制御	146
7. 3. 3.	その他のシステム制御	146
7章4節	要素制御	147
7. 4. 1.	エンジン制御	147
7. 4. 2.	モーター制御	147
7. 4. 3.	走行性能制御	147
7. 4. 4.	バッテリー制御	147
7. 4. 5.	動力合成機構制御	148
7章5節	既存制御技術の理解	149
7. 5. 1.	燃料噴射制御システム	149
7. 5. 2.	燃料噴射量の決定	150
7. 5. 3.	噴射時期の決定	151
7. 5. 4.	燃料噴射制御の難しさ	152
7. 5. 5.	マイコンプログラム上のエンジン制御の核心	153
8章	センサー	155
8章1節	自動車用センサーの技術的ニーズ	155
8. 1. 1.	環境の変化と制御システムの対応	155
8. 1. 2.	センサーの技術的ニーズ	156
8. 1. 3.	検出対称の拡大と複合化	157
8. 1. 4.	センサー材料の変化	157
8. 1. 5.	作動原理・変換機能の多様化	157
8. 1. 6.	信号処理のインテリジェント化	158

8章2節	自動車用センサーの分類	160
8. 2. 1.	パワートレイン制御システムのセンサー	160
8. 2. 2.	車両制御システムのセンサー	162
8. 2. 3.	ボディー制御システムのセンサー	162
8. 2. 4.	情報通信システム系のセンサー	163
8章3節	GPS センサーの簡単な実験	167
8. 3. 1.	GPS センサーの信号	167
8. 3. 2.	NMEA0183 フォーマット	168
8. 3. 3.	マイコンの選定	168
8. 3. 4.	センサーとマイコンの接続	169
8. 3. 5.	プログラム	170
8. 3. 6.	完成した GPS ロガー	173
8. 3. 7.	GPS ログを地図上に表示	174
8章4節	この章のまとめ	176
9章	モニタリング表示	177
9章1節	モニタリングシステム概要	177
9章2節	求められる ECU の処理	179
9章3節	デジタルメータの開発	181
9. 3. 1.	表示器の選択	181
9. 3. 2.	マイコンの選択	182
9. 3. 3.	VR（可変抵抗器）	183
9. 3. 4.	照度センサー	183
9. 3. 5.	温度センサー	186
9. 3. 6.	配線	187
9. 3. 7.	プログラム	187
9. 3. 8.	動作の様子	196
9. 3. 9.	シリアル通信によるシステムの拡張	197

1章 スマートハウス

1章1節 「スマートハウス」とは？

大手ハウスメーカーは、2011年相次いで市場に「スマートハウス」を投入しました。スマートハウスとは、IT（情報技術）を使って家庭内のエネルギー消費が最適に制御された住宅。具体的には、太陽光発電システムや蓄電池などのエネルギー機器、家電、住宅機器などをコントロールし、エネルギーマネジメントを行うことで、CO2排出の削減を実現する賢い省エネ住宅のことを指します。省エネ・創エネ設備を備えた住宅がエコ住宅であるのに対し、エネルギーマネジメントシステム（HEMS: Home Energy Management System）で最適化された賢いエコ住宅がスマートハウスという認識です。

経済産業省は・・・従来、「スマート＝IT」が一般論であり、3.11以降は特に「スマート＝節電（新エネ）」という意識が強くなっていますが、「スマート」に、「快適」、「健康」、「断熱」、「クール・ジャパン」、「リフォームやリサイクルのしやすさ」等の本来住宅があるべき概念や、これからの住宅が有するべき概念を含ませたものです・・・としています。

(2012年1月 経済産業省 情報経済課)

スマートハウスは、次のような設備を備えています。

(1) HEMS (Home Energy Management System)

ヘムスと読みます。「エネルギーの見える化」と「エネルギーの制御」を可能にする設備です。HEMSで節電への意識が高まるだけでなく、従来の住宅では不可能だった外部からの家電製品の制御もできるようになります。ITの助けを借りて総合的に省エネを図ります。

(2) スマートメーター

HEMSの中核となる設備です。電力供給者（電力会社）が各家庭や建物の消費電力をリアルタイムで把握するための設備です。スマートグリッドに欠かせない存在です。HEMSと協調して省エネを図ります。

(3) 省エネ家電

従来の家電製品よりも省エネ性能が高い家電製品のことです。

(4) 太陽光発電

屋根や壁面に太陽電池パネルを設置して、太陽光を利用して発電を行います。

(5) 家庭用蓄電池

電力をストックしておくための設備です。イメージ的には携帯電話やデジタルカメラの電池の巨大版です。

(6) コージェネレーション

ガスを用いて発電を行い、その発電時に生まれる熱を給湯や冷暖房などに活用する設備のことです。

(7) 電気自動車

電気の力で動く自動車のことです。家庭用蓄電池として活用することもできます。

(8) ハイブリッドカー

電気で動くモーターとガソリンで動くエンジンの両方を備えている自動車のことです。電気自動車と比べると環境性能は劣りますが、走行距離は優れています。

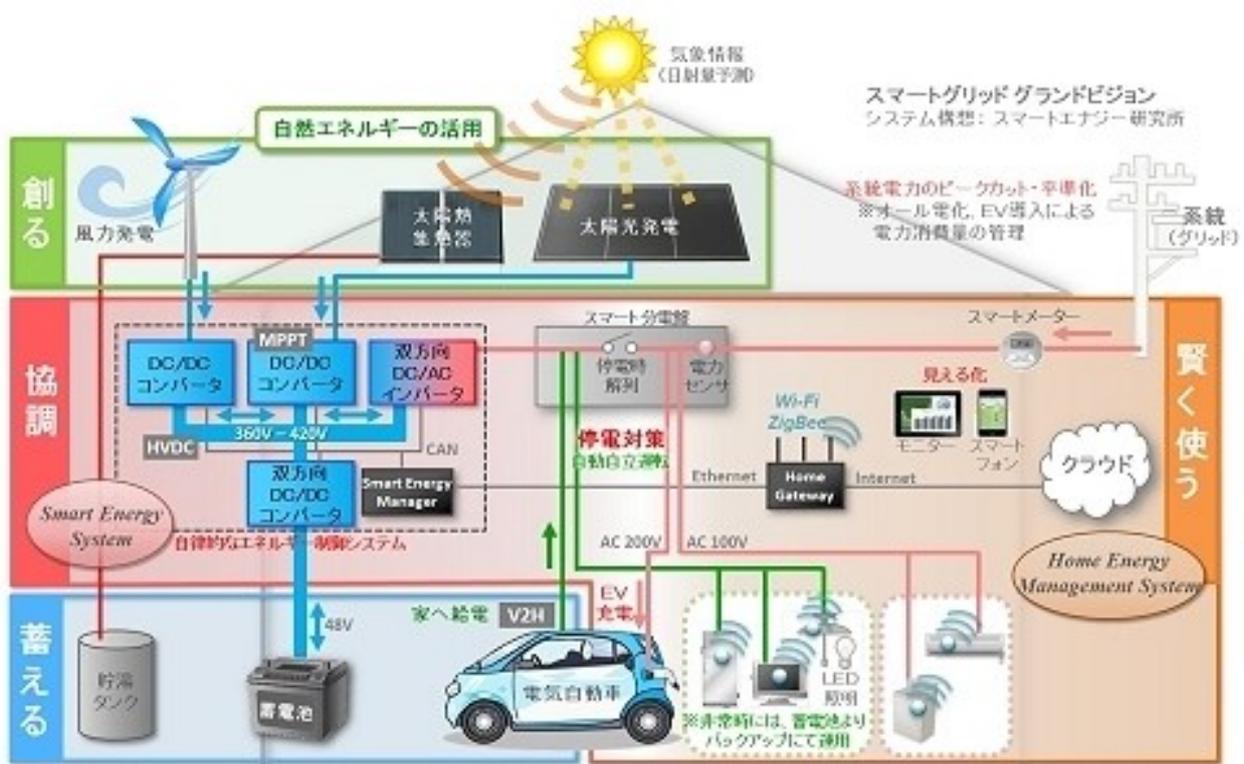


図 1 スマートハウスの仕組み (出典: 福岡スマートハウスコンソーシアム)

1章2節 創エネ機器

太陽電池や燃料電池を使い電力を作り出す機器を創エネ機器と呼びます。

1.2.1. 電池

電池を分類すると図のようになります。

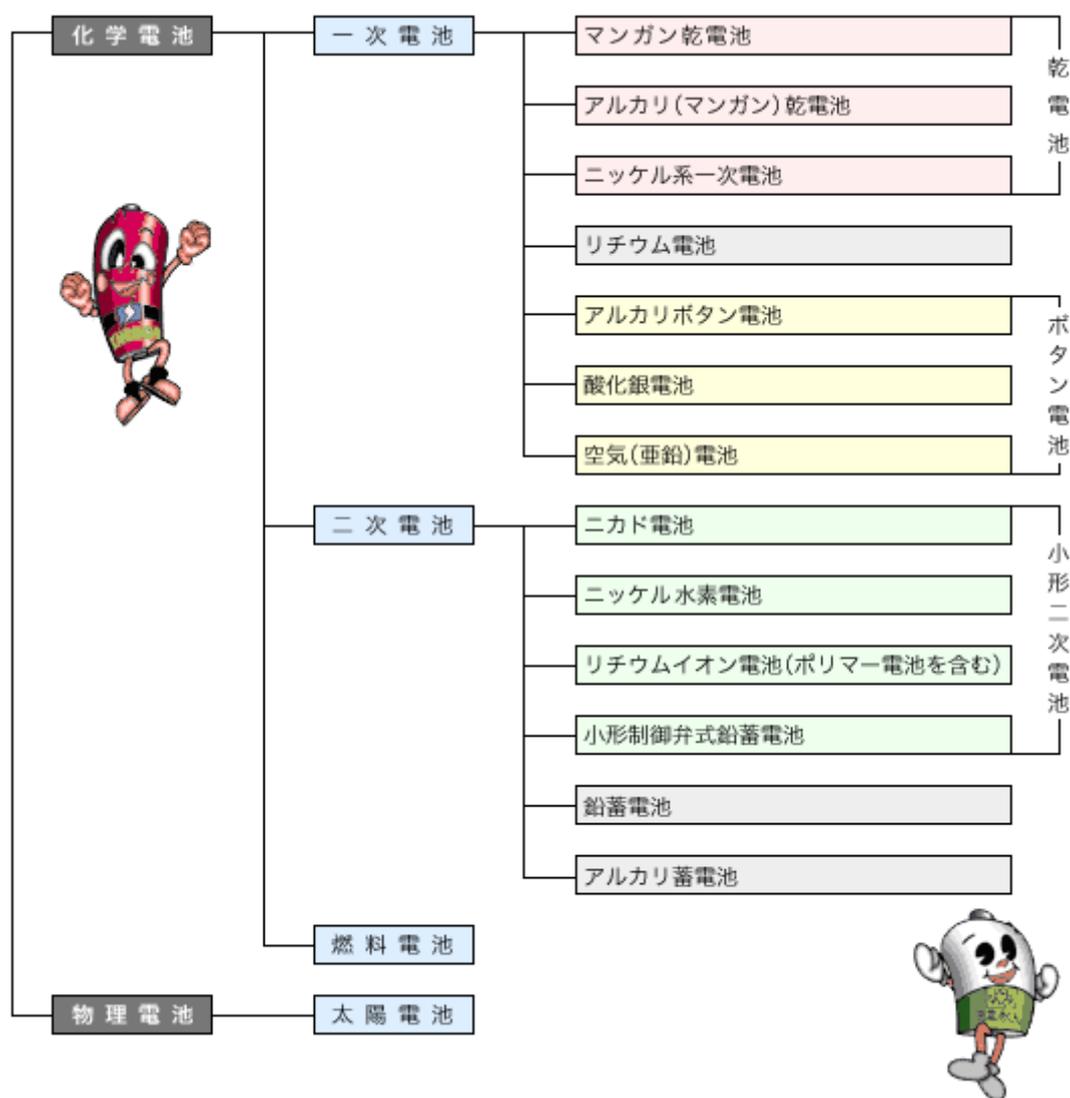


図 2 電池の種類 (出典：日本電池工業会)

化学電池は、物質の化学反応を利用して電気エネルギーを作り出すもので、一般に使われる乾電池などもこの化学電池に含まれます。化学電池のうち、充電のできない使い切りのものを一次電池、充電が可能なものを二次電池とよびます。

燃料電池は、水素を利用した化学反応で、電気エネルギーを取り出すものです。

物理電池は、太陽光や熱などで外部から得られる再生可能エネルギーを電気エネルギーに変化する電池です。

1. 2. 2. 太陽電池

太陽電池は、太陽の光エネルギーを電気エネルギーに直接変換するものです。電気エネルギーへの変換過程では、CO₂などの温暖化ガスを排出せずにクリーンなエネルギーを供給し続けてくれます。その寿命は今後 50 億年とも言われています。詳細は、2 章（太陽光発電システム）を参照してください。

1. 2. 3. 燃料電池

水素と酸素を化学反応させて発電する電池です。水を電気分解する過程を逆にして利用する仕組みで電気を取り出しています。発電の結果、電気エネルギーと水ができますが、二酸化炭素などは排出されず、クリーンなエネルギー源とされています。詳細は、3 章（燃料電池）を参照してください。

1. 2. 4. ガス発電・給湯暖房システム

ガスを燃料とするガスエンジンで発電を行い、その際に発生する熱を回収し家庭の給湯・暖房に利用するシステムで「エコウィル」という名称が使われています。



図 3 エコウィル（出典：東京ガス）

1章3節 蓄エネ機器

スマートハウスで使われている、電気エネルギーを蓄える蓄電池には、鉛蓄電池とリチウムイオン蓄電池があります。鉛蓄電池は価格も安く、長く利用されていて、鉛のリサイクルも確立しており、安心して利用できる機器ですが、エネルギー密度が低いので、大きな電力を蓄電するには、重量と容積が大きくなり、不便な面があります

リチウムイオン蓄電池は、エネルギー密度が高く容積が小さくできるので、電子機器によく利用されています。最近では、電気自動車（EV）やプラグインハイブリッド自動車（PHV）、燃料電池車（FCV）の蓄電池として利用されるようになりました

自動車のリチウムイオン蓄電池に蓄電して、停電時や電力消費のピーク時に自動車から電気を取り出して家庭で利用しようとするV2H（Vehicle to Home）という利用方法も実用化を迎えています

詳細は、4章（蓄電池）を参照してください

熱エネルギーの保存する蓄エネもあります。エコキュートと呼ばれ、市販されています。ヒートポンプを使い、大気中の熱エネルギーを取り出して、熱交換器を通じて温水タンクに熱として蓄エネする仕組みです。わずかな電気で、効率の良い運転をすることで、蓄エネもしますが、省エネにもなります

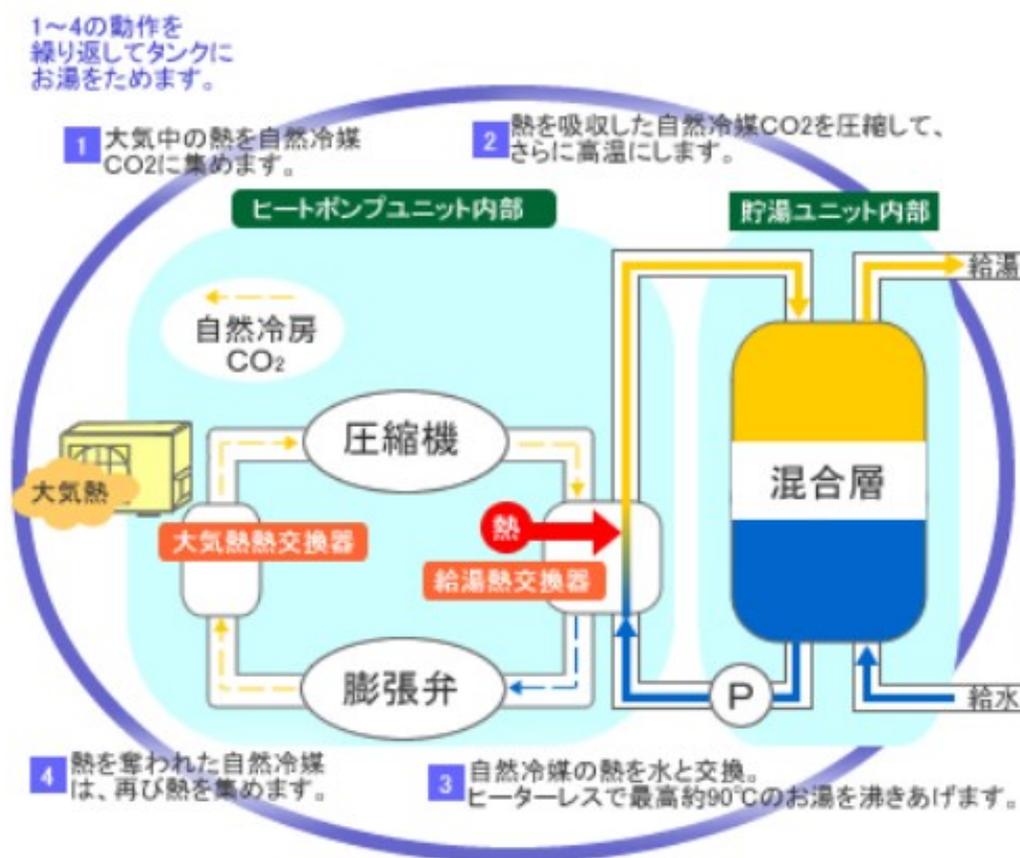


図 4 エコキュート（出典：セキスイハイム）

1章4節 省エネ機器

エネルギー消費量の少ない電化製品や、HEMSのように直接的にはエネルギーを省かないけれど、間接的に省エネに役立つといったような機器も含まれます。インターネット経由でHEMSと接続し、すべての機器が連携して省エネを進めることができます。

1. 4. 1. トップランナー制度

日本では、経済産業省・資源エネルギー庁が中心となって、トップランナー制度を設けて省エネを推進しています。

トップランナー制度とは、該当する機器のエネルギー消費が最も優れている製品の値を判断基準値として、省エネルギー効果を評価するものです。1999年に制度化されて以来、継続的に取組改善が行われ、エネルギー消費効率の改善は表のように評価されています。

表 1 エネルギー消費効率の改善（出典：経済産業省）

機器名	エネルギー消費効率の改善 (実績)	エネルギー消費効率の改善 (見込み)
テレビジョン受信機 (ブラウン管テレビ)	25.7% (1997→2003年度)	16.4%
ビデオテープレコーダー	73.6% (1997→2003年度)	58.7%
エアコンディショナー※ (ルームエアコン)	67.8% (1997→2004冷凍年度)	66.1%
電気冷蔵庫	55.2% (1998→2004年度)	30.5%
電気冷凍庫	29.6% (1998→2004年度)	22.9%
ガソリン乗用自動車※	22.8% (1995→2005年度)	22.8% (1995→2010年度)
ディーゼル貨物自動車※	21.7% (1995→2005年度)	6.5%
自動販売機	37.3% (2000→2005年度)	33.9%
電子計算機	99.1% (1997→2005年度)	83.0%
磁気ディスク装置	98.2% (1997→2005年度)	78.0%
蛍光灯器具※	35.6% (1997→2005年度)	16.6%

※ エネルギー消費効率の指標が、COP、燃費又は効率 (lm/W) であることから、エネルギー消費量の削減効果としては、逆数であることに留意。

注 ディーゼル乗用車は、出荷台数が激減したことから一概に比較できないため掲載していない。

また、現在は家電品だけでなく建築資材や産業機械などの品目にも制度を広げています。

1. 4. 2. 省エネラベリング制度

省エネラベリング制度は、製品に貼付した省エネルギーラベルで、消費者に対して家電製品の省エネ性能に関する情報提供を行うことにより、省エネ効果の高い製品の普及促進を目的として、2000年にスタートした制度です。

省エネ基準を達成した機器は緑色のeマーク、達成していない機器にはオレンジ色のeマークで表示をしています。

1. 4. 3. 統一省エネルギーラベル

統一省エネルギーラベルは、市販している製品を相対評価で表示し、比較が容易な情報を提供するために多段階評価による表示を実施しています。他方、市販品の性能差が小さい機器については、多段階評価のない簡易版ラベルを用いて表示しています。



(参考) 統一省エネルギーラベル

The unified energy label for a TV shows a 5-star rating, energy efficiency percentage (100%), annual energy consumption (450kWh/year), and an estimated annual electricity cost of 9,900 yen. The label is titled 'この商品の省エネ性能は?' (What is the energy efficiency performance of this product?).

【多段階評価制度】
 ・省エネ性能を5つ星から1つ星の5段階で表示。市場における製品の性能の高い順に5つ星から1つ星で示している。
 ・トップランナー基準を達成しているものがいくつの星以上であるかを明確にするため、星の下に矢印でトップランナー基準達成・未達成の位置を明示。

【省エネラベリング制度】
 ・トップランナー基準の達成のものには緑色のeマーク、未達成のものにはオレンジ色のeマークを表示。
 ・その他トップランナー基準の達成率及びエネルギー消費効率(年間消費電力量等)を表示。

【年間の目安目安料金】
 ・エネルギー消費効率(年間消費電力量等)等を分かりやすく表示するために年間の目安電気料金を表示。

統一省エネラベル

図 5 ラベリング制度と統一省エネルギーラベル (出典：経済産業省)

1章5節 HEMS

HEMS (Home Energy Management System : ヘムス) とは HAN (Home Area Network) と連動し家庭内のエネルギー管理をするためのシステムです。HEMS により電力の使用を効率化でき、節電や CO2 削減に役立ちます。HEMS とはエコロジーと居住快適性を実現するための技術です。

HEMS を構築する際にソーラーパネルを使用して太陽光発電や燃料電池で電気を起こし、その電気をリチウムイオンバッテリーのような蓄電池に蓄えて使用する場合、発電量と蓄電量をリアルタイムで把握する必要があります。更に HEMS は家電機器が現在どれくらい電気を消費しているかを知る必要もあります。個々の家電機器の電力消費量を把握し制御することによってより細かな電力管理が可能になります。電気自動車のバッテリーの残量の把握や充電のタイミングを制御する必要もあります。HEMS では電気自動車のバッテリーを電源として利用するというアイデアもあります。

経済産業省が HEMS 導入の促進のために助成金制度を設ける等、HEMS が注目を集めています。現状の HEMS は家電の消費電力を可視化（見える化）する機能が主体で一部に制御機能を有しています。更に HEMS はスマートハウスの一部として連動し省エネ性や利便性の向上を実現します。

HEMS 機器は相互接続性を確保することで複数メーカーの家電製品でシステムを構成することが可能になり、より利便性が向上します。よって、HEMS の大きな課題は家電のコントロールの規格化です。現在、HEMS の規格統一に様々な活動が行われています。

スマートハウスを実現するための HEMS の通信プロトコルは欧米では SEP2.0 (Smart Energy Profile 2.0) という規格を推進していますが、日本では ECHONET-Light (エコーネットライト) という規格を独自に策定しました。

日本の HEMS は、ECHONET-Light による標準インターフェースを持つことが要件の 1 つとなっています。

1章6節 スマートメーター

スマートメーターとは、双方向通信機能を持つ電力やガスなどの計量機器のことで、遠隔検針や遠隔開閉（送電停止・再開を遠隔で行う）など、電力会社などとメーター間のデータ送受信を行う機能を持つものを指しますが、この機能に加えて、家庭内の機器とも双方向通信し、機器制御ができるものを指すこともあります。

現在の家庭内の電力メーターは、累積の消費量を計測する機能しかないものがほとんどです。検針員が月に一度、メーターの数値を目視で確認し、前月との差を計算して請求するため、今どのくらい使っているかなどを需要家が知ることができません。スマートグリッドでは、効率よく需給バランスをとるために、需要側と供給側の情報を双方向で管理・制御することが重要とされ、その要となる機器の一つがスマートメーターです。

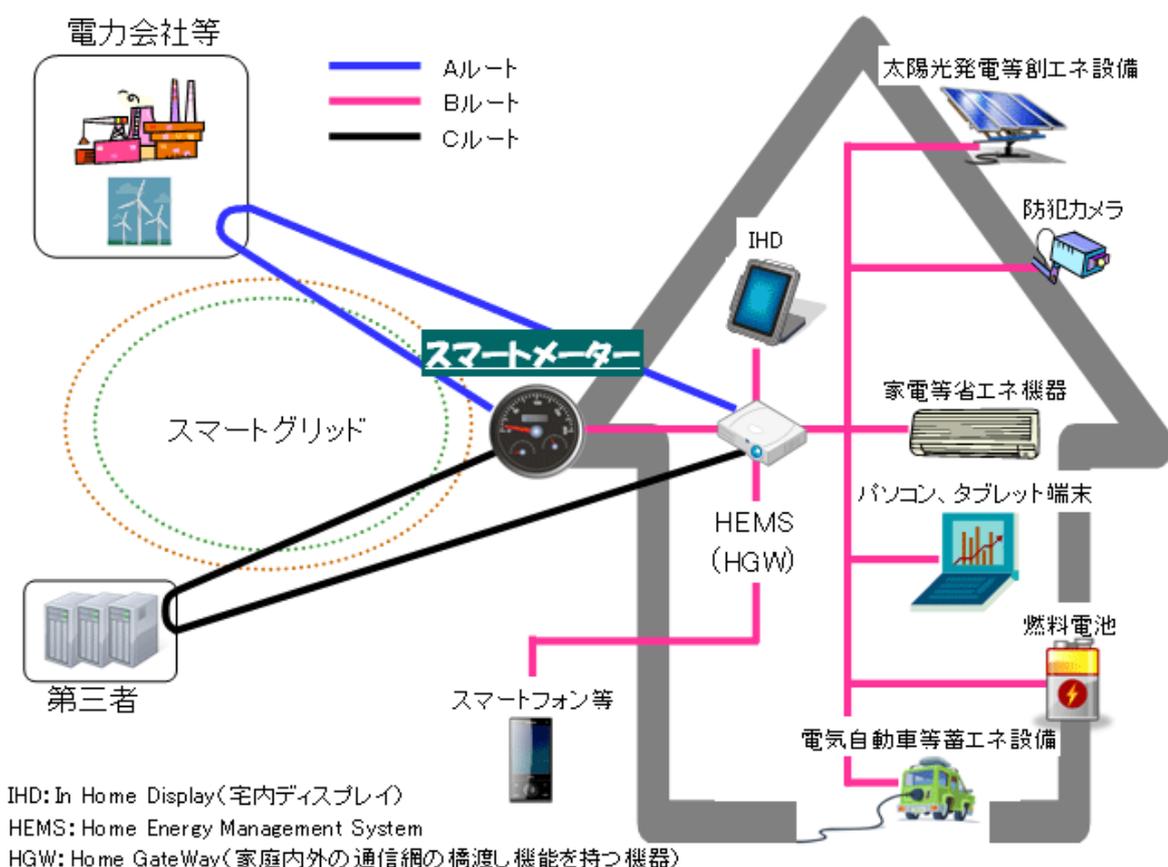


図 6 スマートメーターと情報通信経路（出典：大和総研）

日本では、経済産業省に設置された「スマートメーター制度検討会」で、スマートメーターによる電力情報の取得方法が検討され、三つの情報通信ルートが示されました。

◇A ルート：電力会社の持つネットワークか Web を使って、スマートメーターで取得したデータを電力会社へ送るルート。電力会社が、ここで取得したデータやデータを加工・分析した情報を、家庭に提供します。

◇B ルート：スマートメーターと HEMS (Home Energy Management System) をつなぎ、

パソコン、専用の表示端末（IHD：In Home Display）、スマートフォンなどで管理・制御します。HEMSは家電、太陽光発電、電気自動車などの「省エネ・創エネ・蓄エネ」機器・設備を管理し、機器・設備ごとの電力量監視やオン／オフなどを行うことが想定されています。

◇C ルート：データセンターを持つIT企業のような第三者が、電力会社経由やWebを通じてデータを取得します。



図 7 スマートメーターの例（出典：東芝）

1章7節 スマートグリッド

スマートグリッドとは、ネットワークによる通信や制御の機能を取り入れ、電力供給の最適化を図る次世代電力網のことです。既存の電力網では外部との通信や制御はできませんが、スマートグリッドでは ICT 活用により、リアルタイムで各家庭やビルやマンションなどの電力使用量を把握することができ、過負荷による停電防止や送電調整に役立ちます。

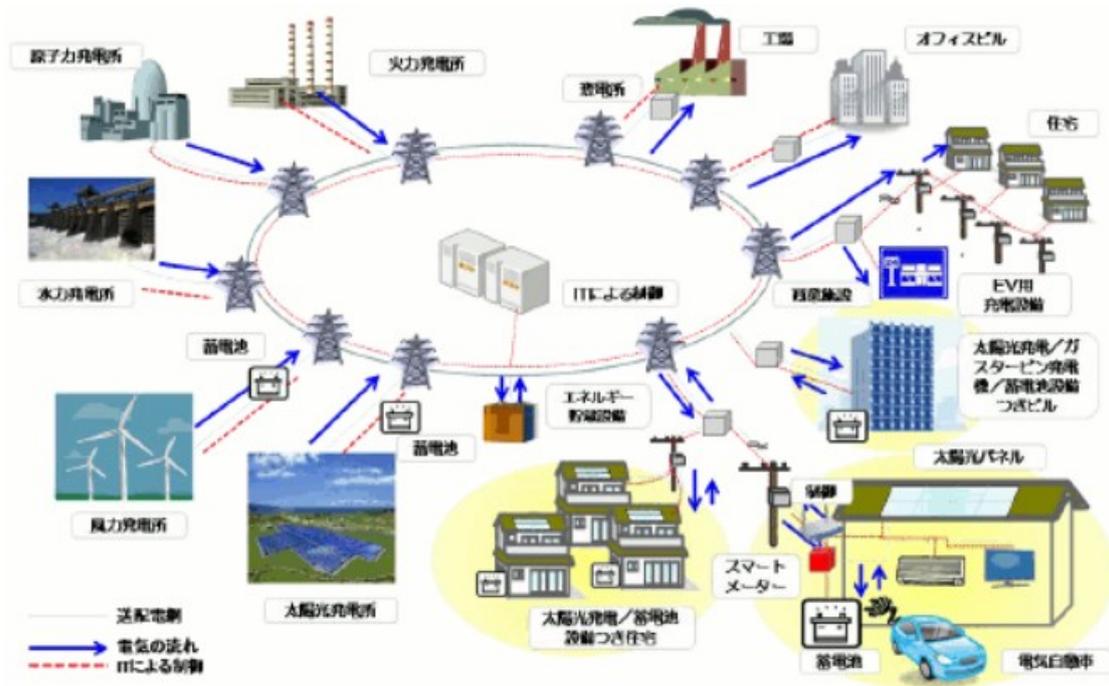


図 8 スマートグリッドのイメージ (出典：経済産業省)

1章8節 スマートシティ

化石燃料や原子力などのメガインフラが中枢をなし、街や工場などの需要側に流し込む形での送配電システムが取られてきたものを、需要側でも最大限に再生可能エネルギーを取り込み、メガインフラと需要側の双方向で管理する電力系統制御を行っていく。これがスマートシティです。

例えば、各住宅の屋根に太陽電池を設置して発電し、生活に必要な電力を賅ったうえで余剰分は電気自動車に蓄電します。各家電と電力メーターに ICT（情報通信技術）を組み込んで、家電を外からでもリモートコントロールできるようにします。こうして『スマートハウス』が誕生します。スマートハウス同士が連携して互いに電力を融通し合うことで『スマートコミュニティ』ができあがります。コミュニティで余った電力は消費量の多い都市部に環境負荷の軽いグリーン電力として送れるようになります。このように双方向配電システムで電力系統のインテリジェント化を実現し、再生可能エネルギーを最大限に利用する社会が『スマートシティ』なのです。

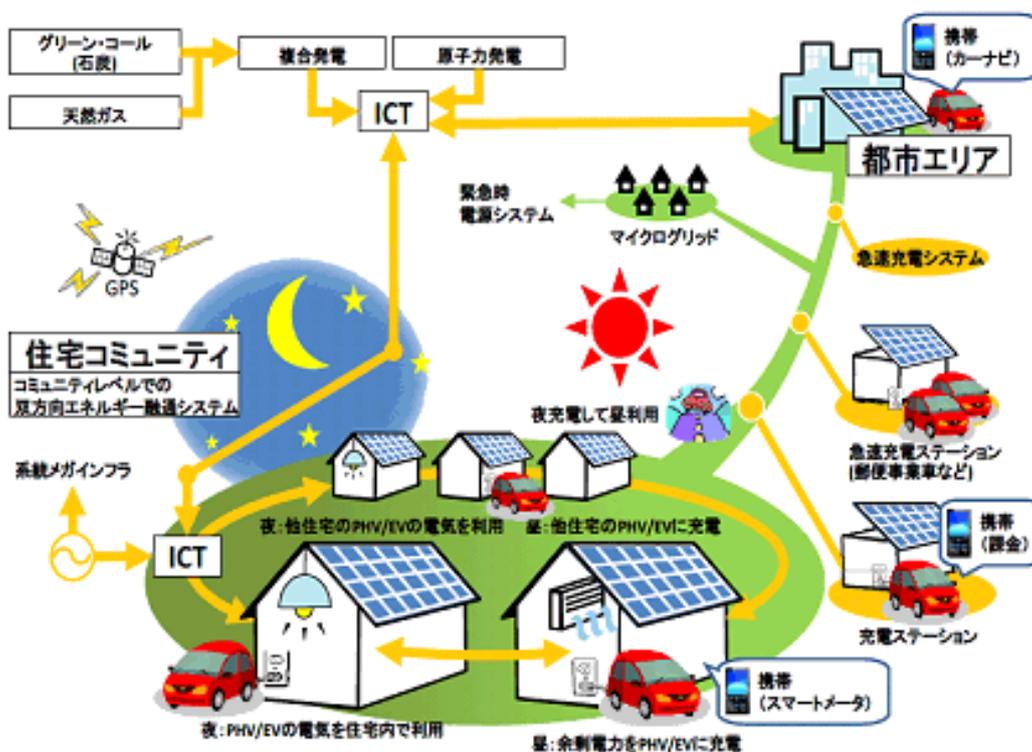


図 9 スマートシティ概念図 (出典：国土交通省)

スマートコミュニティのイメージ

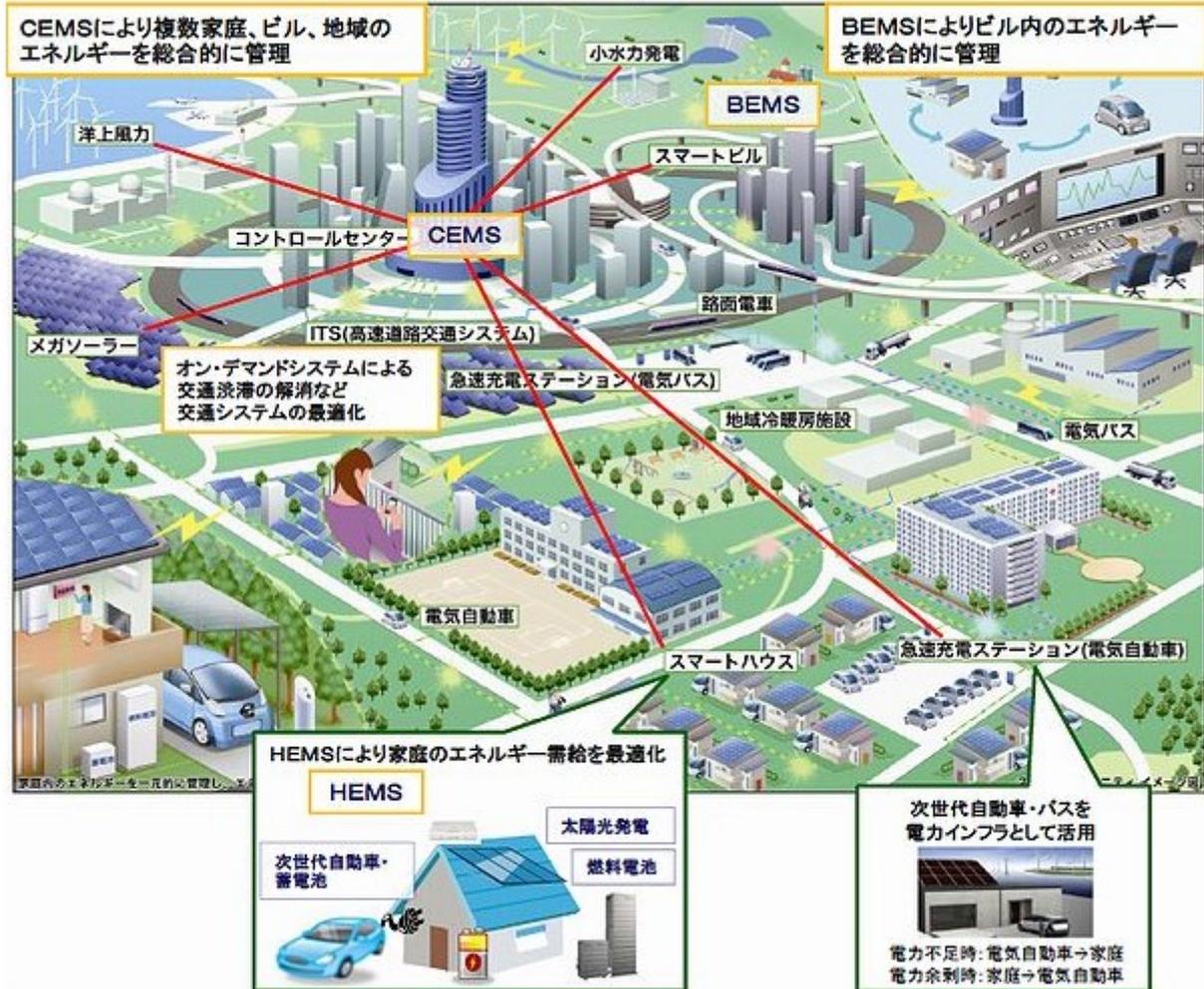


図 10 スマートコミュニティのイメージ (出典：経済産業省)

スマートシティにある一般住宅は HEMS によってエネルギーを管理していますが、他の施設のエネルギー管理をおこなうことも必要です。

◇BEMS: HEMS が一般住宅 (Home) を対象としているのに対して、ビル (Building) を対象とするエネルギー管理システムのことを言います。一般住宅よりもビルの方がエネルギーをたくさん消費するので、HEMS よりも省エネ効果は高いでしょう。

BEMS の管理対象となるのは、主に配電・空調・換気・照明・OA 機器などといった設備です。ビル内に太陽光発電システムや蓄電池やコージェネレーションなどといった各システムが導入されている場合は、これらもネットワークに加えることによって管理の対象に含めることが可能です。

現実にはまだ蓄電システムやコージェネレーションシステムを導入しているビルが数少ないため、ほとんどの BEMS は電力使用量の可視化が主な目

的となっています。ただ、一部のビルでは **BEMS** を導入して、各種電気設備の制御も行われています。

◇**FEMS** : **FEMS** とは「**Factory Energy Management System**」を省略した言葉で、「フェムス」と読みます。

HEMS は家庭、**BEMS** はビルのエネルギーを管理するシステムですが、こちらは **Factory** ですので工場のエネルギーを管理するシステムとなります。配電・空調・換気・照明・製造ラインなどといった設備が管理の対象です。

◇**SMES** : スマートハウスに対応するスマートストアのエネルギー管理システムを言います。スマートストアというのは、エネルギーを賢く管理することで、環境性と経済性を向上させた店舗のことです。

◇**CEMS** : **CEMS** の「**C**」は、「**Community**」の頭文字です。つまり、**CEMS** とは地域やコミュニティのエネルギーを管理するシステムのことです。対象地域内の発電所・工場・ビル・家庭のエネルギーを総合的に管理するシステムとなりますので、スマートグリッドに欠かせない要素と言えます。

2章 太陽光発電システム

2章1節 地球上の環境エネルギー

地球上に到達する太陽光のエネルギー量は 1m^2 当たり約 1kW 。もしも地球全体に降り注ぐ太陽エネルギーを 100%変換できるとしたら、世界の年間消費エネルギーを、わずか1時間でまかなうことができるほど巨大なエネルギーであり、しかも、枯渇する心配がありません。現在、日本は、石油や石炭などのエネルギー資源のほとんどを諸外国からの輸入に頼っていますが、こうした化石燃料は使い続けられればいずれなくなってしまうものです。太陽の光という無尽蔵のエネルギーを活用する太陽光発電は、年々深刻化するエネルギー資源問題の有力な解決策の一つです。また、クリーンであることも大きな特長です。発電の際に地球温暖化の原因とされている二酸化炭素 (CO_2) も発電時にはまったく排出しません。エネルギー源の確保が簡単で、地球にもやさしい太陽光発電では、日本が世界でもトップクラスの太陽光発電技術を有する国でもあります。

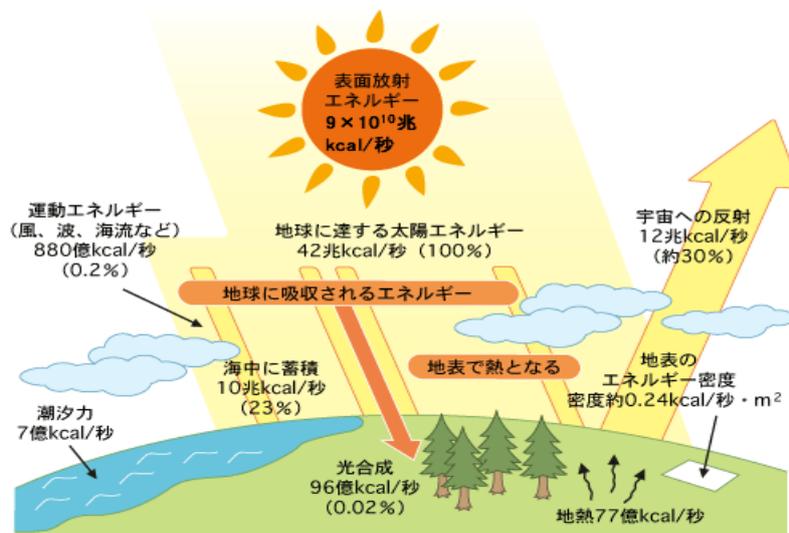


図 11 地球上の環境エネルギー (出典：太陽光発電協会)

種類	エネルギー量
水 力	5 億 Kcal/秒
潮 汐 力	7 億 Kcal/秒
地 熱	7 7 億 Kcal/秒
風 波	8 8 0 億 Kcal/秒
太 陽 光	4 2 0, 0 0 0 億 Kcal/秒

図 12 (出典：太陽光発電協会)

2章2節 太陽光発電の原理

太陽光発電は、太陽電池で発電した電気を利用するものです。では太陽電池はどのような仕組みで、発電するのでしょうか。

太陽電池は電子に光エネルギーを吸収させて、エネルギーを持った電子を外部的に取り出します。エネルギーを持った電子を取り出す仕組みには、「半導体」を使います。

この半導体には N 型半導体と、P 型半導体の 2 種類があります。太陽電池は基本的に、この N 型と P 型の半導体を積み重ねた構造をしています。(右図)

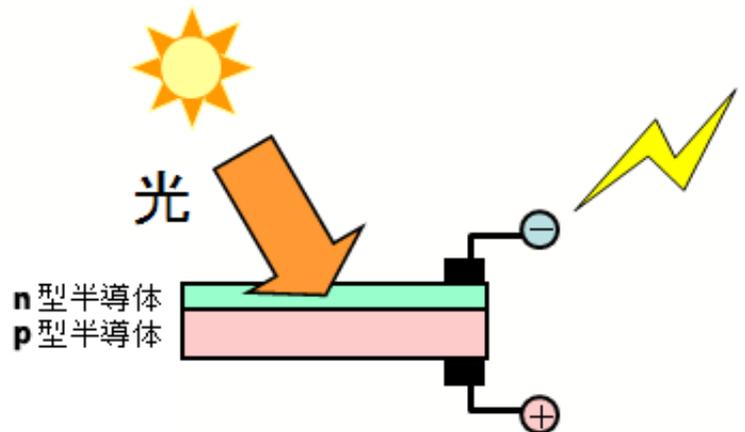


図 13 (出典：産業技術総合研究所)

N 型の半導体は“動きやすい”電子(伝導電子)がやや多く、接触した材料に電子が逃げ出しやすくなっています。逆に P 型の半導体は伝導電子がやや少なめで、電子が足りない場所(正孔)を持っています。この 2 つを接合すると、N 型半導体から P 型半導体へと伝導電子が逃げ出して、正孔と打ち消し合います。

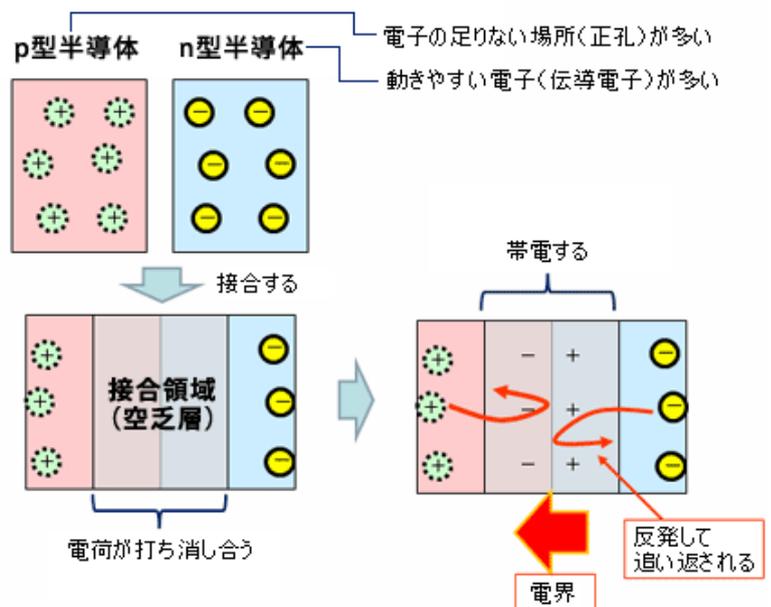


図 14 (出典：産業技術総合研究所)

電子が逃げ出した後の N 型半導体は電子が足りなくなりますので、プラスに帯電します。同様に、余分に電子をもらった P 型半導体はマイナスに帯電します。このために接合部分に電界(内部電界)が生じます。内部電界は、N 型半導体から逃げ出そうとする電子の流れを妨げるように働き、N 型から P 型へ電子が流れようとする力と釣り合った所で安定します。接合部分では電子と正孔が結びついた状態で動けなくなっていますが、そこには常に内部電界が働いています。伝導電子があれば、電界によって N 型半導体へと押し流される状態になっています。(上図)

そこで接合部分の半導体に光が当たると、光のエネルギーによって新たに伝導電子と正孔が「叩き出され」ます。内部電界に導かれて、伝導電子はN型半導体へ、正孔はP型半導体へと移動します。その結果、電子を外部へ押し出す力「起電力」が生まれます。起電力は光を当てている間持続し、次々に電子が押し出されることで、外部の電気回路に電力が供給されます。押し出された電子は外部の電気回路を通じてP型半導体の側へ戻り、正孔と結合します。
 (右図)

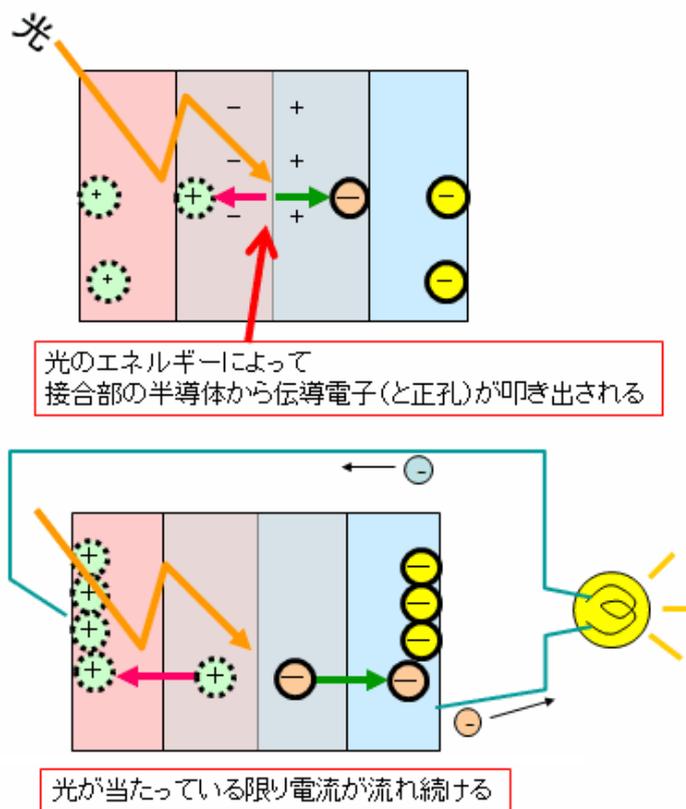


図 15 (出典：産業技術総合研究所)

2章3節 太陽電池の材料

太陽電池の材料では3つの種類に大別できます。

① シリコン系

太陽電池にシリコンを用いるタイプのものです。「結晶シリコン型」と「アモルファスシリコン型」に分けることができ、どちらも更に細かく複数の種類に分類されています。最も広く使われています。

② 化合物系

ガリウムやカドミウムや銅などを用いるタイプのものです。用いる原材料によって細かく種類が分けられていますが、量産化されている「CIS系太陽電池」が代表的存在です。

③ 有機系

有機化合物を用いるタイプのものです。常温で塗布するだけで製造でき、カラフルで軽量なものも造れる太陽電池です。寿命と変換効率の向上が課題ですが、屋内用のものは量産が始まっています。大幅な低コスト化が見込めるため、期待がかけられています。

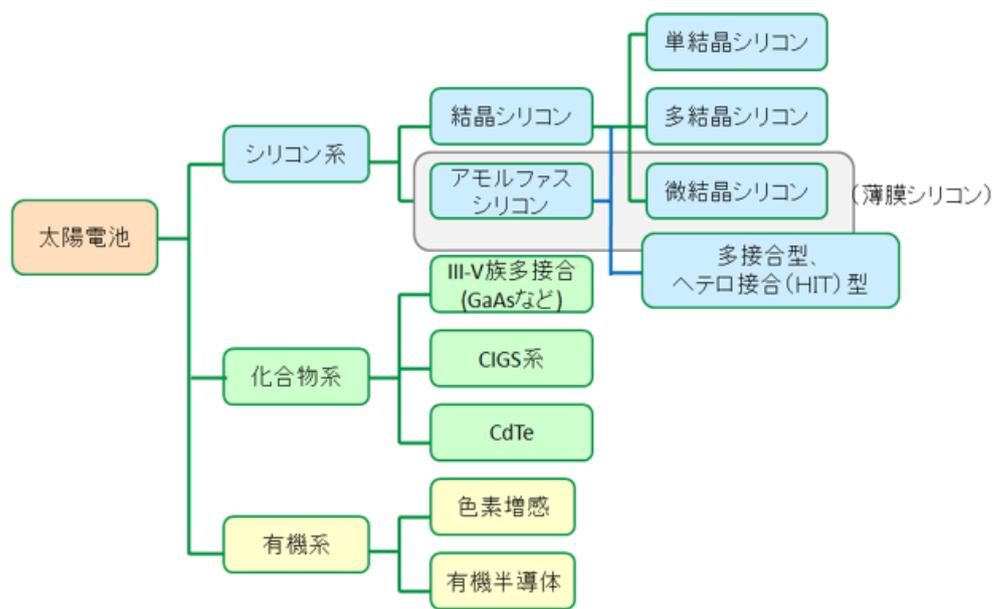


図 16 材料による太陽電池の分類 (出典：産業技術総合研究所)

建物の屋根や壁面などに太陽電池パネルを設置することによって発電を行うという発電方法です。国の政策や精度、各種助成などで、最近では太陽電池パネルを搭載している一般住宅が増えてきています。

2章4節 太陽電池の構成

太陽電池として使用できる最小の単位を「セル」と呼び、セルをつなぎ合わせ、ガラスやプラスチックで表面保護し、アルミフレームなどで強度を持たせて、取扱いしやすくしたものを太陽電池「パネル」もしくは太陽電池「モジュール」と呼んでいます。太陽電池モジュールをさらに複数枚並べて接続した1系統を「ストリング」と呼び、このストリングをさらに複数接続したものが「アレイ」です。住宅の屋根等に設置されているものはストリングを複数並べた「アレイ」と呼ばれるものです。

セル→モジュール（パネル）→ストリング→アレイ→屋根

モジュール単体の電圧が低いので、一般的にはモジュールを直列接続して電圧を上げてストリングとして使用しています。アモルファスや化合物半導体などの薄膜電池は、単体の電圧が高いので、モジュールを並列接続するのが一般的です。

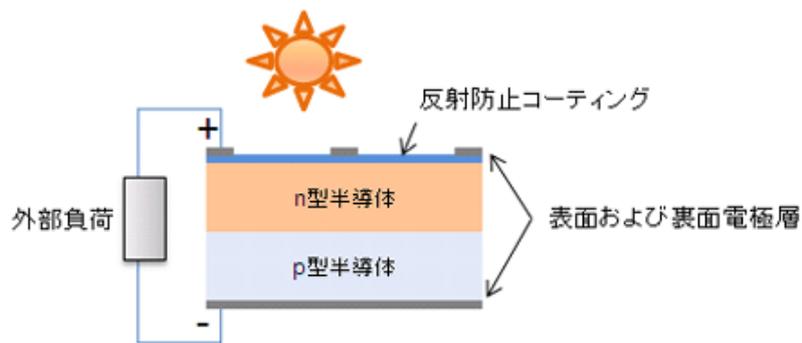


図 17 太陽電池セルの構造・断面（出典：テキサスインスツルメンツ）

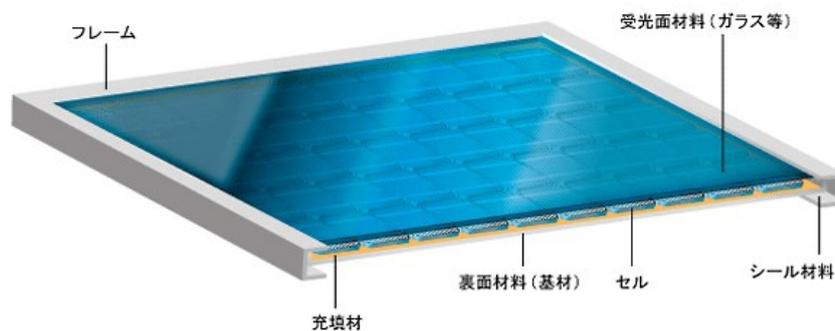


図 18 太陽光モジュールの構造（出典：日立ハイテック）

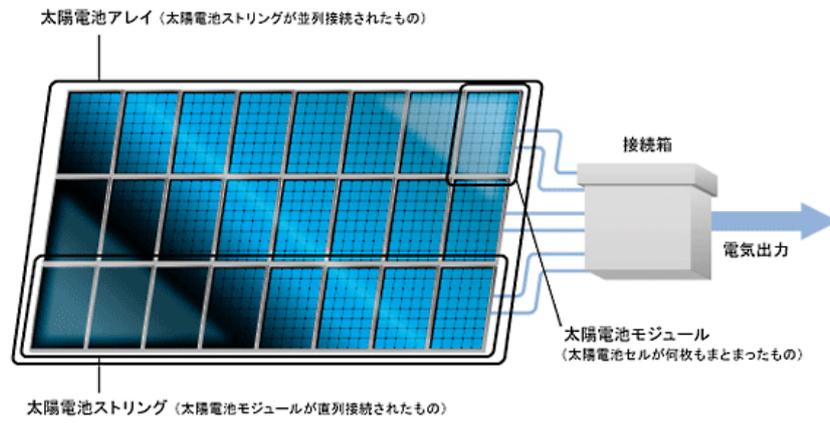


図 19 太陽電池アレイの構造 (出典：日立ハイテック)

2章5節 太陽電池の特性

2. 5. 1. 出力特性

一般に太陽電池の出力は図に示すように、発生する電圧と電流の相関関係曲線で表されます。

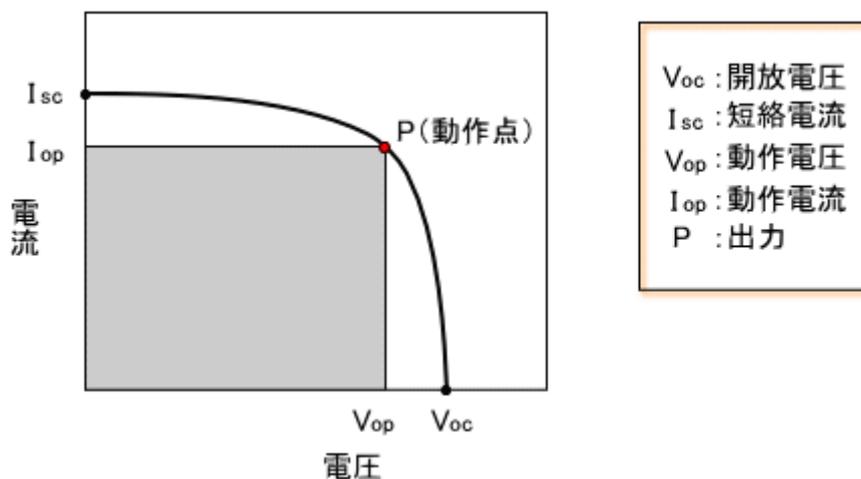


図 20 太陽電池の出力特性 （出典：日本イーテック）

- ①解放電圧（Voc）：太陽電池に何もつながらない状態での両端に発生する電圧値。
- ②短絡電流（Isc）：太陽電池の両端を短絡させた状態での電流値。
- ③動作点（P）：太陽電池から出力を取り出すために設定された電圧に対して発生する電流が決まります。このときの電圧、電流の点を動作点といいます。

※太陽電池は、一次電池のように電圧を保って出力する定電圧電源と異なり、電流を一定に保とうとする定電流電源として動作するので、上の図のような出力特性になります。

2. 5. 2. 最大出力点

太陽電池の出力は I_{op} と V_{op} と原点を結ぶ面積（上図のグレー部分）で示されます。太陽電池を効率よく使うということは、グレー部分の面積を最大にする I_{op} と V_{op} を設定する必要があります。また、太陽電池の出力が最高となる動作点を最適動作点（このときの出力を最大出力： P_{max} 、電圧を最大動作電圧： V_{pm} 、電流を最大動作電流： I_{pm} ）と呼びます。

2. 5. 3. 温度特性

太陽電池は外気温や日照によりモジュール温度が上昇すると発電電圧が下がる（出力特性の P 点が左に移動する）特性を持っています。その低下率は太陽電池の物性により異なりますが、結晶系では 1℃温度が上昇すると約 0.4%低下します。そのため太陽電池を設置するときはできるだけ温度が上昇しないように配慮しなければなりません。

具体的には、太陽電池アレイの背面と設置面の間に空間を持たせて、空冷効果を得るなどの施工がなされます。

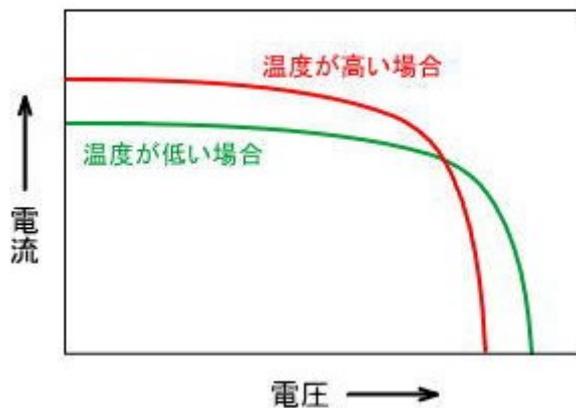


図 21 太陽電池の温度特性 （出典：日本イーテック）

2. 5. 4. 放射照度特性

受光面の放射照度（日射強度）が変化すると、その強さに比例して発生電流（短絡電流）が変化し、それに伴い出力電力も変化します。放射強度は天候に大きく依存します。また、太陽電池の設置方向や設置角度により、受光面での放射強度も変わります。

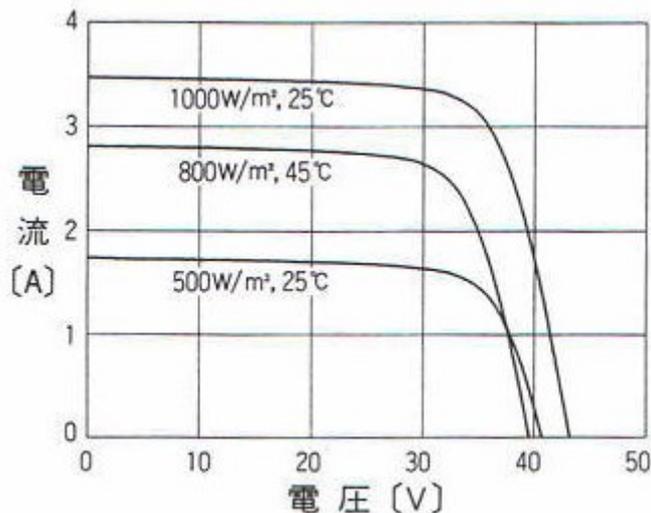
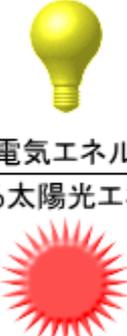


図 22 太陽電池の放射強度特性 （出典：日本イーテック）

2. 5. 5. 変換効率

太陽電池の性能は、太陽光から電気への変換効率としてあらわされます。変換効率は、単位面積（1 m²）あたりに注がれる光エネルギーのうちどれほどの割合で電気エネルギーが取り出せるかという値になっています。



$$\text{変換効率} = \frac{\text{出力電気エネルギー}}{\text{入射する太陽光エネルギー}} \times 100(\%)$$

図 23 変換効率 （出典：日本イーテック）

入射する太陽光エネルギー＝モジュール面積（m²）×1,000（W/m²）

出力電気エネルギー＝モジュール公称最大出力（W）

表 2 太陽電池の変換効率 （出典：『太陽電池とその応用』：パワー社）

	変換効率、%
単結晶Si	24
多結晶Si	17
アモルファスSi	12
単結晶Ga-As	29
単結晶In-P	19
CdS/CsTe	12
CdS/CsInSe ₂	14

上表に素材別の太陽電池の変換効率を示します。ここで分かるように太陽電池の変換効率は、おおよそ 10~30%未満です。入手の容易な多結晶 Si が素材の場合は 17%です。シリコン系太陽電池の変換効率は 35%が理論上の限界とされています。

一方、2001 年度に開始された「新エネルギー技術研究開発」プロジェクト（NEDO）において、研究用の非集光セルではありますが、エネルギー変換効率 35.8%を達成し、太陽

電池の世界記録を樹立しました。さらに、2011年には、自社記録を更新する36.9%を達成しています。現在は、2030年までにモジュール変換効率40%を達成する目標に向かい、研究開発が続けられています。加えて、レンズなどを利用して太陽光を集光し50%を超えるエネルギー変換効率を目指す「集光型太陽光発電システム」の実用化開発にも取り組んでいます。また、「量子ドット型太陽電池」により、2020年までに効率40%を目指す。(東京大学 先端科学技術研究センター) という研究も進んでいます。

さらに多方面での技術開発が進めば、高効率太陽電池を搭載した充電フリーの次世代自動車なども現実になってくるのではないのでしょうか。

2章6節 太陽光電池の効率的運用

実際の太陽光発電では、MPPT (Maximum Power Point Tracking) 装置という、太陽電池の特性に合わせた運用により、効率的な運用がはかられています。次の図のような定格が 24V1A の出力特性を持っている太陽電池パネルを例にして考えてみましょう。

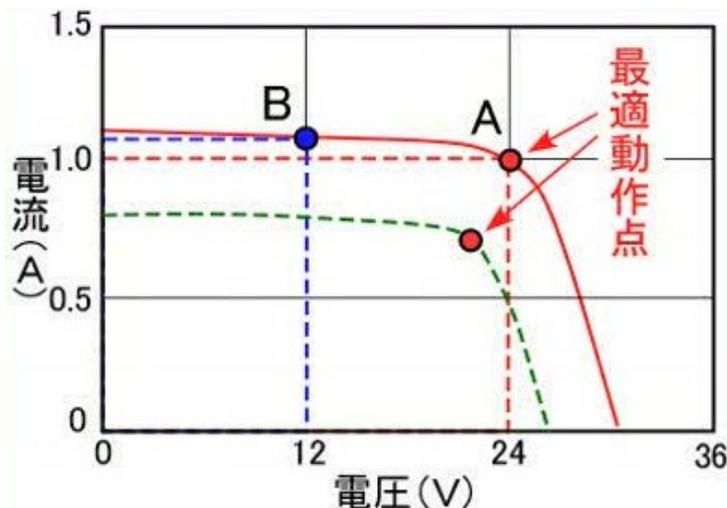


図 24 太陽電池パネルの出力特性 (最適動作点 24V1A を仮定)

太陽電池パネルの定格が 24V1A ということは、一番効率よく動作する状態 (図の A 「最適動作点」の状態) で、最大 $24(\text{V}) \times 1(\text{A}) = 24(\text{W})$ を発電できるということです

ところがここに 12(V) のバッテリーを直結すると、電圧が 12(V) に引き下げられてしまう (図の B 点の状態で作動する) ので、取り出せる電流は約 1.1(A) になりますが、出力は $12 \times 1.1 = 13.2(\text{W})$ しか取り出せず、本来の太陽電池パネル最大性能の約 55% に下げられてしまうことになります。

効率が半分になるので充電時間は理想的な場合の約 2 倍かかると考えて良いでしょう。さらに太陽電池パネルに入る光の量が減ると出力特性も図の内側の点線 (緑色) のように変化して、最適動作点の条件も変化してきます。

さてここで最適な制御 (MPPT 制御) を行う電圧コンバータを太陽電池パネルとバッテリーの間に入れて、余分な電圧を電流に変換して充電すると、太陽電池パネルは、最大動作点の状態の 24V、1A で 24W の電力を発電できます。電圧コンバータの出力は、 $12\text{V} \times 2\text{A} = 24\text{W}$ の電力を充電できるので、発電の効率がたいへんよくなります。

それでは、この状態のまま日射量が減少して、太陽電池が図の内側の点線 (緑色) の条件になった場合は、どうすれば良いでしょうか。

この時の最大電力点は、グラフから 22V、0.7A くらいなので、入力が 22V、0.7A で出力が 12V、1.3A の電圧コンバータを用意すればよいことになります。つまり、最適動作点 (最大電力点) を追従するには、電圧コンバータの入力電圧 (太陽電池の動作点電圧) を変え

てやれば、効率のよい発電ができるようになります。入力電圧を制御できる電圧コンバータは普通にはありませんが、出力電圧を変化させることのできる電圧コンバータは数多く存在します。電圧コンバータは太陽電池パネルを電源として使用した場合には、出力電圧を制御することで入力電圧が制御できてしまうのです。

もう一度、次の図をみて電圧コンバータを使い、蓄電池に充電することを考えましょう。

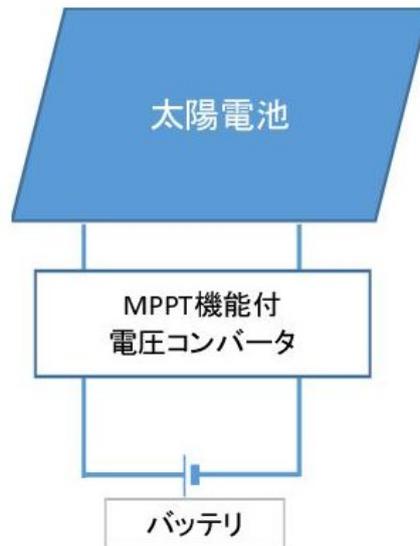


図 25 効率の良い太陽電池駆動

電圧コンバータの出力電圧を意図的に高く設定し、蓄電池との電位差を大きくすると、蓄電池への充電電流が増加します。ということは、電圧コンバータへの入力電流（＝太陽電池から取り出す電流）も増加しますので、電圧コンバータの入力電圧（太陽電池の動作点電圧）を下げる効果が得られます。

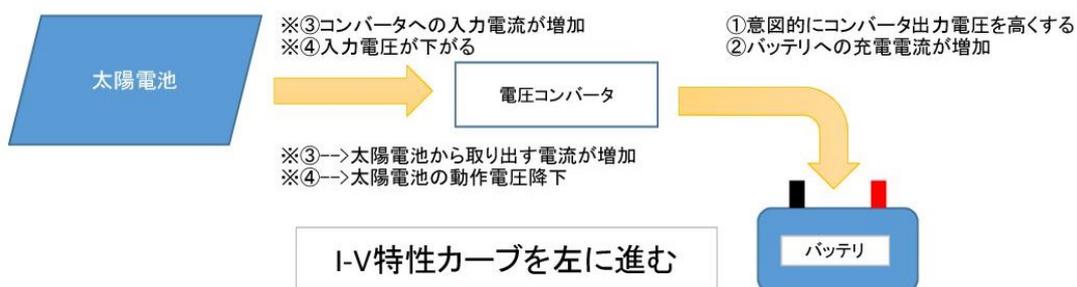


図 26 動作電圧降下

逆に、出力電圧を低く設定すれば、充電電圧と入力電流がともに減少して、電圧コンバータの入力電圧（太陽電池の動作電圧）を上げる効果が得られます。

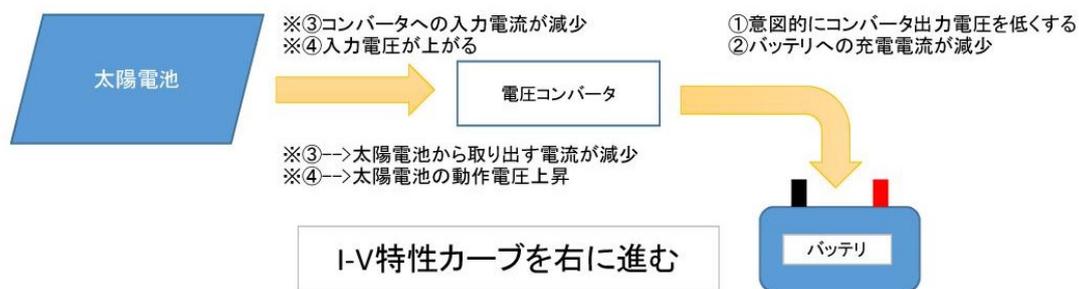


図 27 動作電圧上昇

この際大切なことは、「それぞれの時点での最適動作点がどこにあるかを知ること」です。それには、電圧と電流を測定し $P=V \times I$ の計算結果をもとに、直前の P と逐次比較しながら、動作電圧の上昇か降下かを決めてやればよいのです。この方法を山登り法と呼ばれています。最適動作点電圧がその時の解放電圧の約 80%であることが経験的に分かっているので、電流測定は行わず、動作電圧と解放電圧のみを計測して、I-V カーブを右に進むか（動作電圧上昇）左に進むか（動作電圧降下）決める方法もあり、この方法は電圧追従法と呼ばれています。この方法であれば、システムの簡素化が図れます。

このように、太陽電池の出力特性を最大限に利用して、自動的に太陽電池の最大電力点を追従する制御ができるコンバータを MPPT 装置と言います。

前掲の図 24 のような特性の場合ならば MPPT 制御の有無で太陽電池パネルによるバッテリー充電の効率はおよそ 2 倍違ってくると言えますが、実際にどれくらいの差になるかは、使う機材の実際の特性から判断することとなります。

次に太陽電池パネルと蓄電池、系統電力への実際の接続ブロック図の例を示します。図の点線内に太陽電池パネル側 MPPT と蓄電池側充放電ブロックが含まれています。この例では、系統連携を行うための保護ブロックも含まれています。

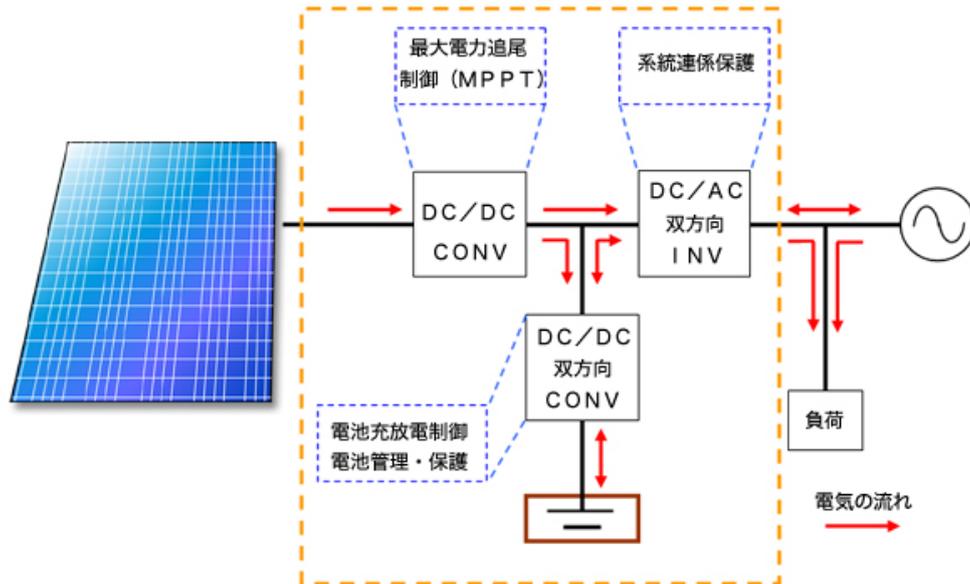


図 28 太陽光パネルと蓄電池側の制御ブロック例 (出典：新電元)

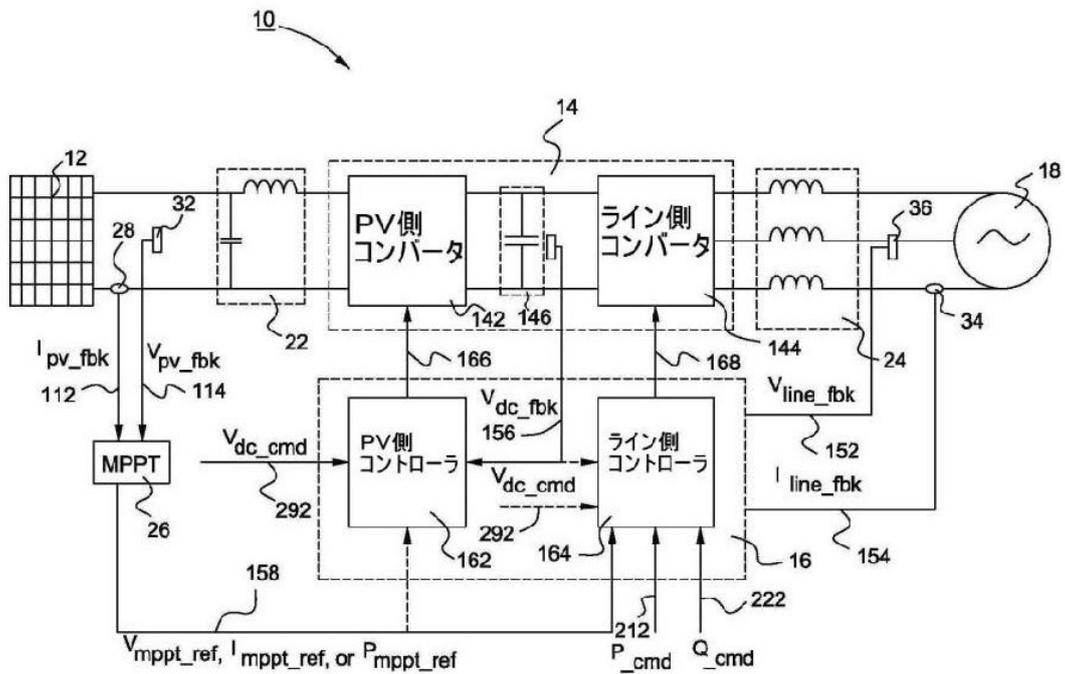


図 29 電力変換システムのための最大電力点追尾およびその方法 (出典：Astamuze)

上の図は、蓄電を行わないシステムのブロック図の例です。MPPT によるコントローラが太陽電池パネル出力側（図では PV 側と表示）とライン側（系統電力側）にコントローラがあり、各々が調停や DC-AC 変換等を行うコンバータに接続されています。

2章7節 住宅用太陽光発電システム

住宅の屋根や壁面に太陽電池モジュールを設置することによって発電し、その電力を自家で利用することによる省エネ推進や、売電して経済的利益を得ることなどを目指しています。国の政策や精度、各種助成などで、最近は太陽電池パネルを搭載している一般住宅が増えてきています。

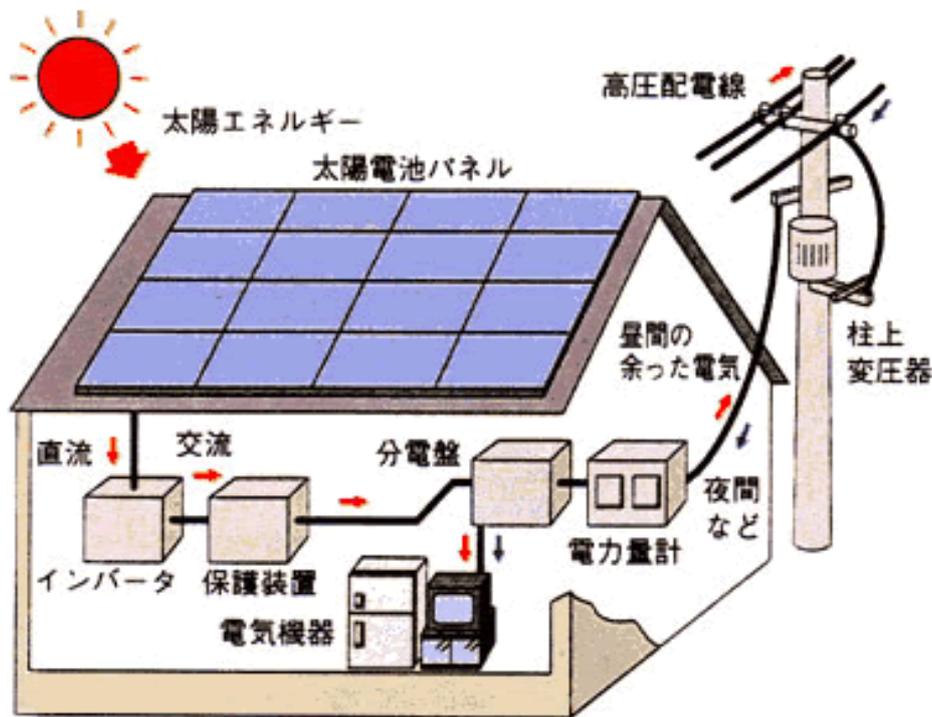


図 30 太陽光発電の概要 (出典：四国電力)

太陽電池パネルは設置場所の面積に応じて、複数枚設置することが一般的です。使用できる太陽電池パネルにも素材別に様々な種類があります。

主な必要設備は、次のようなものになります。

①太陽電池パネル

太陽電池を複数枚並べて接続したパネル状のもので「モジュール」や「ソーラーパネル」と呼びます。

②パワーコンディショナー

太陽電池パネルから生み出した電力は直流電力ですが、それを住宅で使える交流電力に変換するための装置です。「インバータ」と呼ばれることもあります。

③分電盤

住宅内にある各コンセントに電力を送るための装置です。

④蓄電池

太陽光発電による電力やコンセントから得た電力を蓄えるための装置です。

⑤エネルギーモニタ

太陽光発電システムによる発電量や売電量などを確認するためのモニタです。

2. 7. 1. 規模による分類

太陽光発電を規模で分類すると次の二つになります。

①家庭用

一般住宅に使用する比較的小規模のもので「家庭用太陽光発電システム」と表記されることもあります。

②産業用（公共用）

工場やオフィスビル、遊休地などに設置する規模の大きな太陽光発電システムで、出力が1メガワットを超える大規模なものを「メガソーラー」と呼んでいます。最近では、メガソーラーキットまで販売されていて、価格もメガ級で2億～3億円だそうです。



図 31 メガソーラーキット（出典：LOOP）

3章 燃料電池

3章1節 家庭用燃料電池の概要

家庭用燃料電池は、都市ガス・LPガスから取り出した水素と、空気中の酸素を化学反応させて電気と熱を発生させるコージェネレーションシステムで「エネファーム」という統一名称を用いて販売されています。(発電量 200～750W、給湯温度 60～70℃)

利用段階で反応物として水しか排出せずクリーンで、化学反応から電気エネルギーを直接取り出すためエネルギーロスも少ないうえ、電気と熱の両方を有効利用することで、更にエネルギー効率を高めることが可能です。

家庭用燃料電池「エネファーム」とは、「エネルギー」と「ファーム=農場」の造語です。水素と酸素から電気と熱をつくることと、水と大地で農作物をつくることはとても似ています。自分のエネルギーを自分でつくる。これからのエネルギーの考え方を、ファームという世界観により表現し、家庭用燃料電池「エネファーム」は環境にやさしいエネルギーの使い方を実現します。

3. 1. 1. ロゴマーク

人に地球にたくさんの実りをもたらす新エネルギーを表現した木のシンボルマーク。その木の実は、家庭用燃料電池がもたらす「電気」、「お湯」、「快適」、「環境」、「先進」、「未来」などのさまざまな価値を表現しています。



図 32 ロゴマーク (出典：燃料電池普及促進協会)

3. 1. 2. エネファームの仕組み

エネファームの発電原理は、水の電気分解の逆です。都市ガスやLPガスから取り出した水素と空気中の酸素を化学反応させ、電気をつくり出します。発電の際に発生する熱を捨てずにお湯をつくり給湯にも利用する、エネルギーをフルに活用するシステムです。



図 33 エネファームの発電原理（出典：燃料電池普及促進協会）

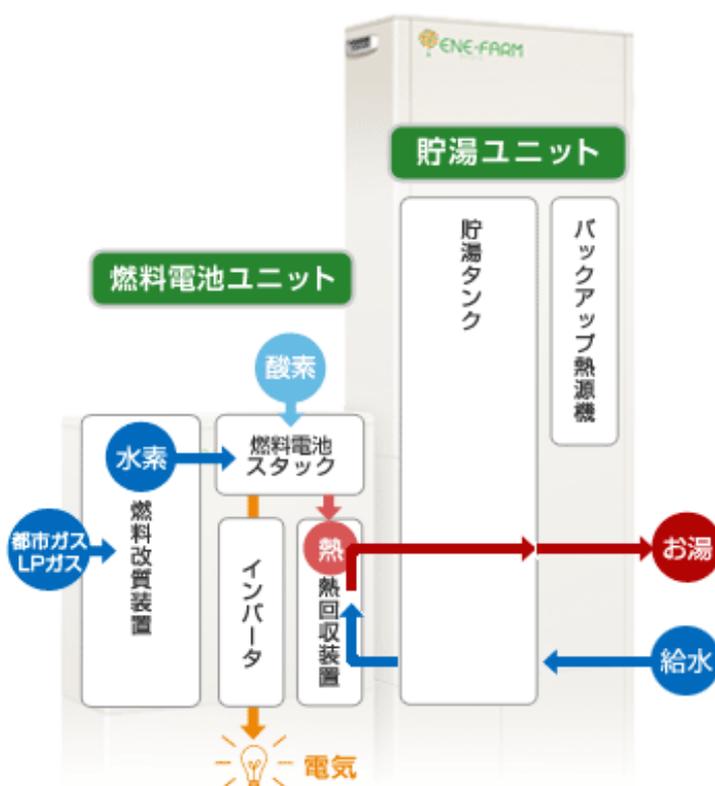


図 34（出典：燃料電池普及促進協会）

エネファームは、**熱主電従**（熱の利用を主に考え、発電については制御を行わない）方式になっています。

3. 1. 3. エネファームのメリット

エネファームの主なメリットは「省エネと電気代削減」です。ガスから水素を取り出し、その水素で燃料電池が発電を行い、同時に排熱を利用して給湯も行うという仕組みで電気代を削減することができます。他にも次のようなメリットがある。

- ◇停電時にも自立運転が可能（一部機種）
- ◇国や自治体の補助金制度が設けられている
- ◇発電時の騒音がない

- ◇エネファーム住宅専用のガス料金優待プランが存在する
- ◇発電時に二酸化炭素や有毒ガスを出さない
- ◇排熱も有効利用できる
- ◇送電ロスがほとんど存在しない
- ◇これまで一戸建て向けでしたが、2014年4月よりマンション向けエネファームが発売されています。(東京ガス)

3. 1. 4. エネファームのデメリット

メリットも沢山ありますが、広く普及することを阻むデメリットもあり、今後の改善が望まれます。

- ◇停電時には自立運転が不可能（一部製品）
- ◇製品価格がまだ高い
- ◇発電した電力を売電することができない
- ◇メンテナンスや点検を行っても寿命は最長で20年ほど
- ◇都市ガスの原料輸入価格が高騰するとガス料金も高騰する可能性がある

3章2節 家庭用燃料電池の種類

燃料電池は、宇宙開発などの特殊な用途をのぞいて次の表のように分類されています。

表 3 燃料電池の種類（出典：燃料電池. net）

	固体高分子形 (PEFC)	リン酸形 (PAFC)	熔融炭酸塩形 (MCFC)	固体酸化物形 (SOFC)
電解質	高分子電解質膜	リン酸	炭酸塩	セラミックス
作動気体	水素	水素	水素 一酸化炭素	水素 一酸化炭素
作動温度	常温～90℃	150～200℃	650～700℃	750～1,000℃
発電効率	30～40%	35～42%	45～60%	45～65%
開発状況	実用化	実用化	研究開発段階	研究開発段階
主な用途	家庭用 携帯機器用 燃料電池車用	工業用 産業用	工業用 分散電源用	工業用 分散電源用

3. 2. 1. 固体高分子形燃料電池（PEFC：Polymer Electrolyte Fuel Cell）

電解質に高分子電解質膜（陽イオン交換膜）を用いているタイプです。小型化と軽量化が可能であることから、家庭用燃料電池や燃料電池自動車などに用いられています。作動温度が低いことや起動が速いなどといったメリットがある一方、他の種類の燃料電池と比べると発電効率は低めとなっています。

(1) 単セルの構造と動作原理

単セルは図に示すように電解質（高分子電解質膜）と燃料極（アノード）、空気極（カソード）という2枚の電極のサンドイッチ構造になっています。

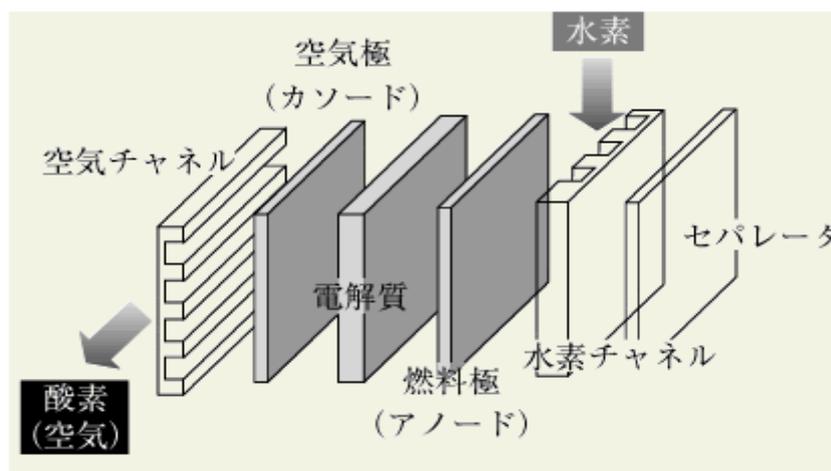


図 35 固体高分子形燃料電池の構造（出典：日本電気技術者協会）

図に動作原理を示します。

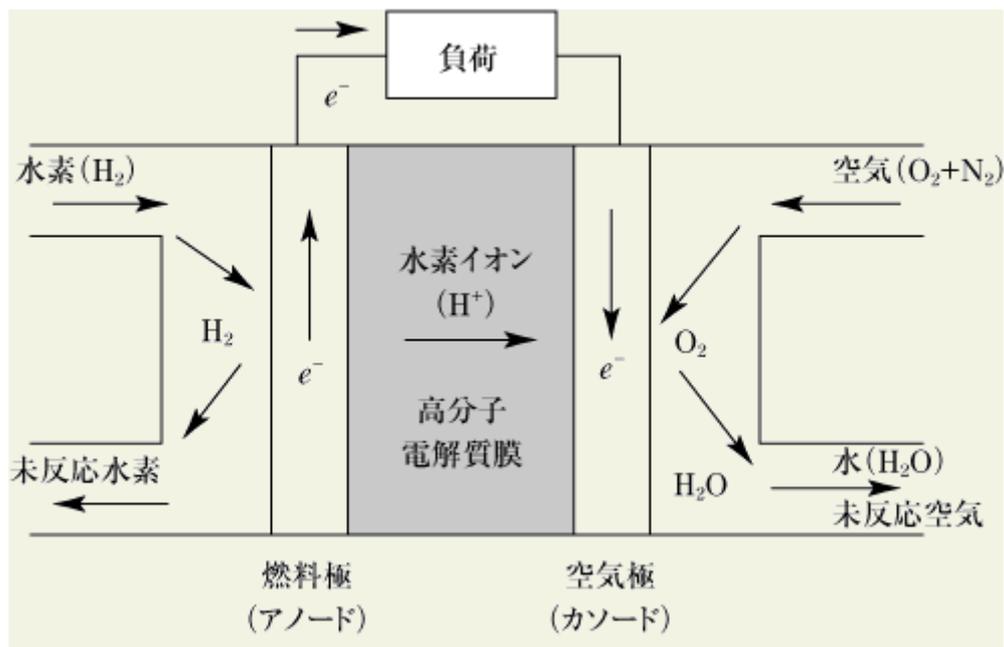
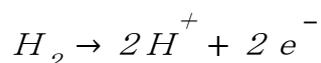


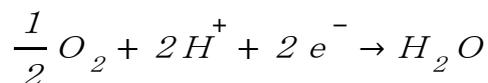
図 36 固体高分子形燃料電池の動作原理（出典：日本電気技術者協会）

燃料極では水素が触媒の白金によって水素イオンと電子に分離されます。

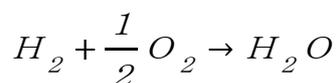


水素イオンは電解質を通して反対極の空気極へ移動し、電子は外部に抜け出し導線を伝って電流となります。

空気極には酸素が導入されます。ここで電解質を通過して入ってきた水素イオンと外部の導線を経由してきた電子との反応で水が生成されます。



両極での反応を総合すると、



という化学式になり、水の電気分解の逆反応が成立していることが分かります。

単セルを挟み込むように配置されたセパレータは、炭素板や導電性の樹脂でできています。水素や酸素はその表面に加工された細かい溝を通して均等に電極へ供給されます。

単セルの電池電圧は通常 0.6~0.8V 程度なので、システム上必要な電圧を得るためには単セルを何段か繰り返し重ねて積層化し、スタックを構成します。

(2) 白金触媒の被毒と燃料の改質

触媒としては白金が使用されます。白金系触媒は、一酸化炭素と非常に結合しやすく、結合した後は活性を失います。これを「触媒の被毒作用」と呼びます。燃料としてメタノール、ガソリンなどから水素を取り出す場合は、一酸化炭素ガス濃度を許容レベル（例えば 10ppm）まで下げるための改質が行われます。改質により被毒作用を抑制します。

3. 2. 2. リン酸形燃料電池（PAFC：Phosphoric Acid Cell）

電解質にリン酸を用いているタイプです。最も古くから研究が進められていたため、前述の PEFC と同様に、既に実用化されています。PEFC よりも発電効率は高いものの、作動温度も高めとなっています。商業用や産業用の比較的規模の大きなコージェネレーションシステムに多く採用されています。

(1) 単セルの構造と動作原理

図に単セルの構造を示します。

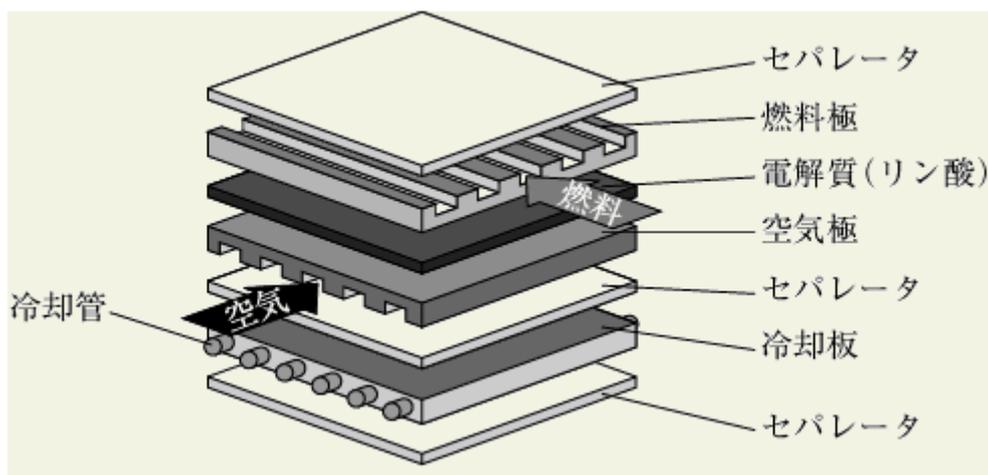


図 37 リン酸形燃料電池の構造（出典：日本電気技術者協会）

燃料極、空気極、電解質が厚さ数ミリメートルの板上に圧縮された構造になっており、燃料極、空気極の二つの電極は、ガスを通しやすいように無数の穴が開いた多孔質の素材で作られています。

動作原理を下図に示します。固体高分子形燃料電池と同様で、燃料極では水素が酸化され水素イオンと電子になり、空気極では空気中の酸素が還元されて水（水蒸気）ができます。

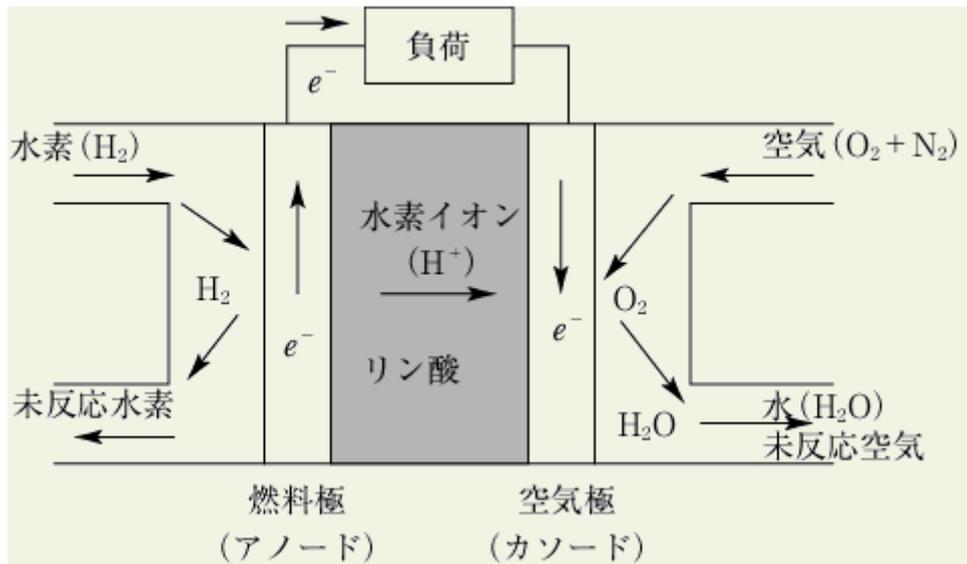


図 38 リン酸形燃料電池の動作原理 (出典：日本電気技術者協会)

(2) 燃料改質器

図に水蒸気改質法による燃料処理システムを示します。白金の被毒対応として、燃料処理によって高純度の水素を電池本体に供給しています。

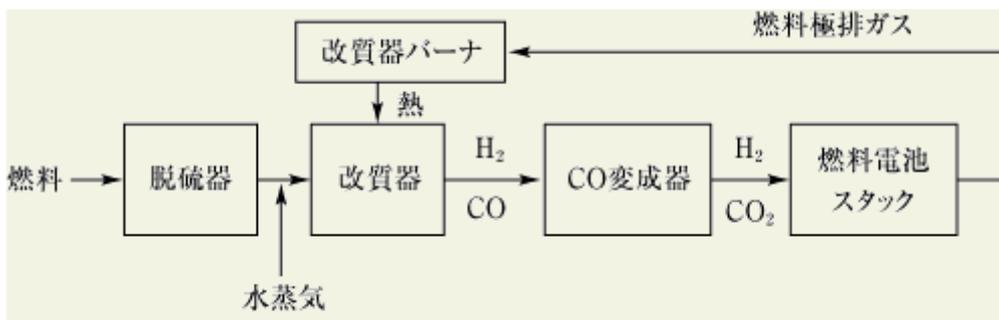
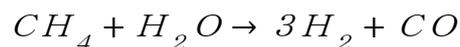


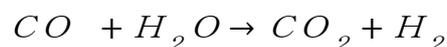
図 39 水蒸気改質法による燃料処理システム (出典：日本電気技術者協会)

燃料処理システムは、脱硫器、改質器、CO 変換器から構成されています。

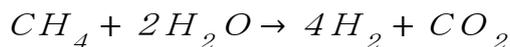
通常天然ガスには微量に硫黄が含まれるので脱硫器によって処理された後、改質器に入ります。水蒸気改質は触媒存在下で水蒸気を投入して処理されます。メタンの場合で示すと、次のよう反応になります。



この反応は吸熱反応で外部から熱を加える必要があり、熱は電池から戻る余剰水素を改質器バーナーで燃焼させることで与えられます。この反応でできた一酸化炭素は、次の CO 変成器でシフト反応と呼ばれる反応によって炭酸ガスに変換されます。



前述の式と合わせた燃料改質全体の反応は次式となります。



3. 2. 3. 熔融炭酸塩形燃料電池 (MCFC : Molten Carbonate Fuel Cell)

電解質に炭酸リチウムや炭酸カリウムなどといった炭酸塩を用いているタイプです。作動温度は高いですが、その代わり発電効率も高くなっています。家庭用として使うことはできませんが、分散電源として活用できるよう実証研究が進められています。

(1) 単セルの構造と動作原理

単セルの構造を図に示します。

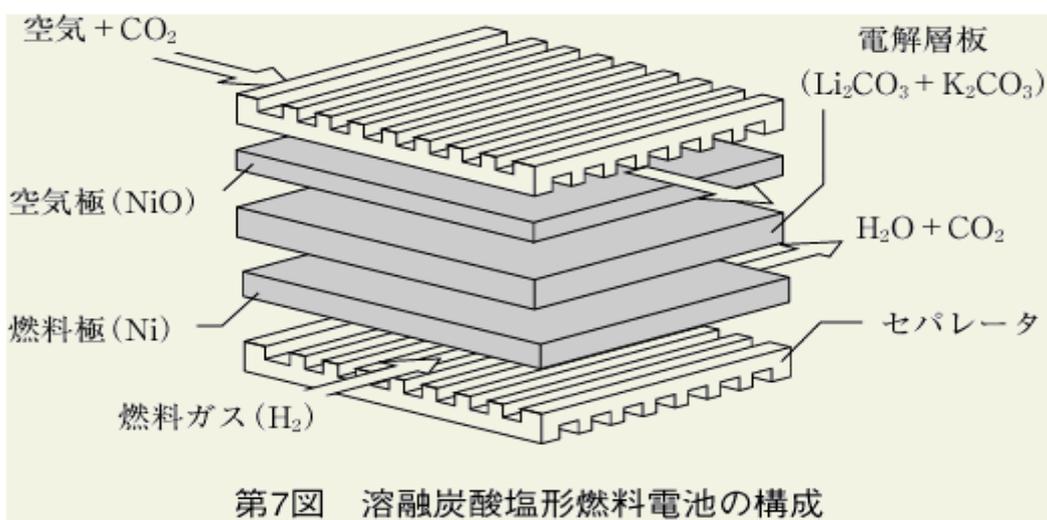


図 40 熔融炭酸塩形燃料電池の構造 (出典：日本電気技術者協会)

単セルは電気伝導性多孔質の燃料極、空気極と熔融炭酸塩を保持する電解質板で構成されています。電解液はアルカリ金属炭酸塩、通常は炭酸リチウム (Li₂CO₃) と炭酸カリウム (K₂CO₃) の混合塩を用います。この混合塩は約 490℃ で熔融状態になり、炭酸イオンの優れた導電体になります。

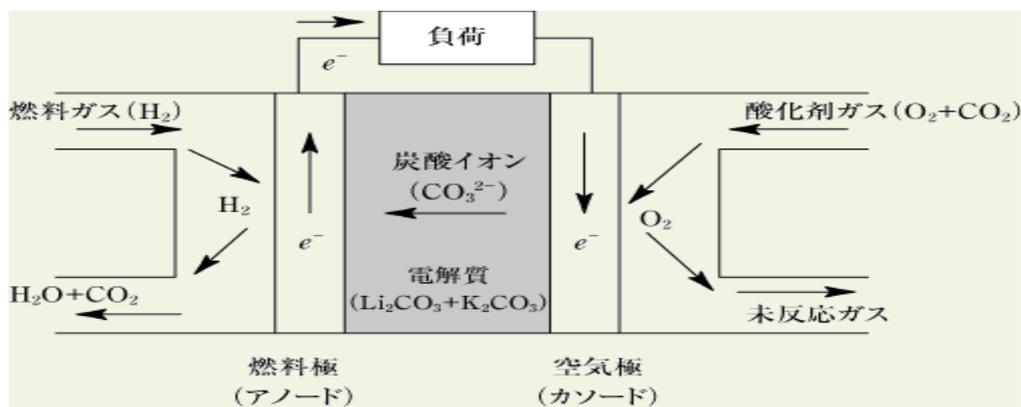
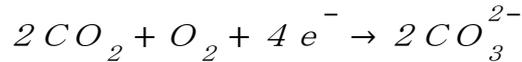
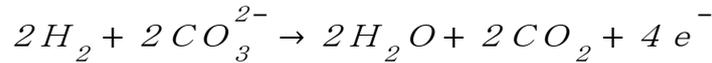


図 41 熔融炭酸塩形燃料電池の動作原理 (出典：日本電気技術者協会)

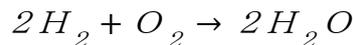
空気極では酸素と炭酸ガスと電子が電解液膜で電気化学的に反応して炭酸イオンとなり、電解質を移動します。



一方、燃料極では水素が電解液を移動してきた炭酸イオンと反応し、水、炭酸ガス、電子を生成します。



全体の反応をまとめると水素と酸素から水を生成する単純な反応です。



炭酸ガスは酸素のキャリアとして働いている。

(2) 外部改質方法と内部改質方法

燃料の改質を行った後の水素を電池に供給する方式を外部改質方式と呼び、この反応を電池の内部で行う方式が内部改質方式です。

内部改質方式は次のようなメリットがある。

- ① 電池の発電に伴う発熱を改質反応に直接利用できる。
- ② システムをコンパクトにできる。
- ③ 改質による吸熱で、電池冷却に必要なエネルギーを節約できる。

しかし、内部改質で副反応的に発生する一酸化炭素が白金を被毒するので、メリットを享受できるのは、高温型燃料電池に限られます。

さらに内部改質方式には改質触媒の位置によって直接方式と間接方式があります。図にこれらの改質方式の構造と動作原理を示します。

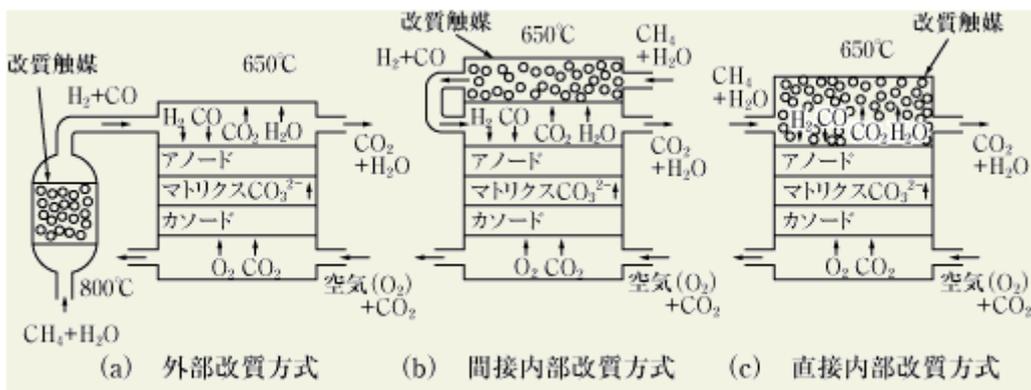


図 42 溶融炭酸塩形燃料電池の改質方式 (出典：日本電気技術者協会)

3. 2. 4. 固体酸化物形燃料電池 (SOFC : Solid Oxide Fuel Cell)

電解質に安定化ジルコニアやランタンなどといったイオン伝導性セラミックスを用いているタイプです。MCFC よりも更に高温で作動する高効率の燃料電池となっています。こちらはまだ研究段階ですが、小型のものであれば既に ENEOS がエネファーム (Type S) に搭載しています。固体酸化物形燃料電池の特長として、電池全体がすべて固体でできており、システム構成が簡単であり、安定した性能と長寿命化が期待できます。また、高温の高品質な排熱を回収できるなどのメリットがあります。

(1) 単セルの構造

すべて固体から構成されることから種々の形状が可能であり、平板型構造と円筒型構造に大別されます。基本的にこれらの単セルがインターコネクタを介して連結する構造になっています。ここで、インターコネクタとはほかの電池のセパレータに対応するもので、固体酸化物形燃料電池ではこのように呼ばれています。

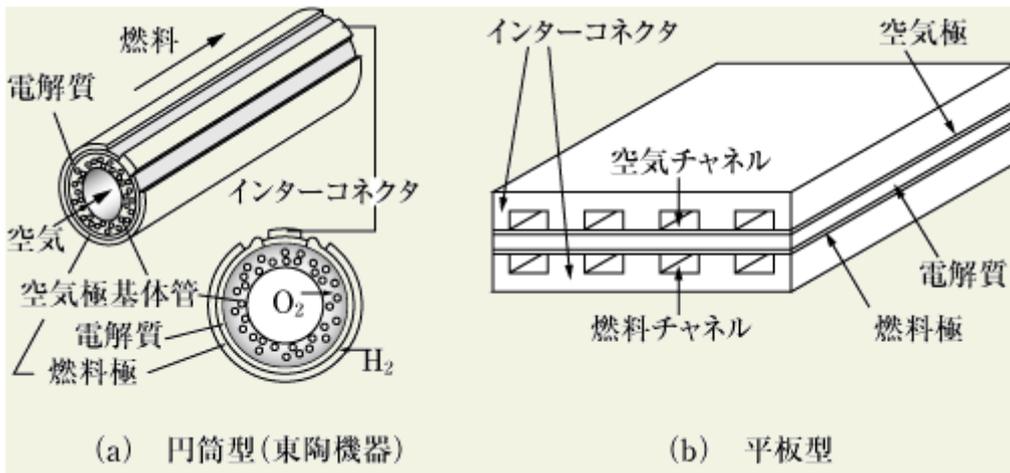


図 43 固体酸化物形燃料電池の構造 (出典：日本電気技術者協会)

(2) 単セルの動作原理

図に単セルの動作原理を示します。

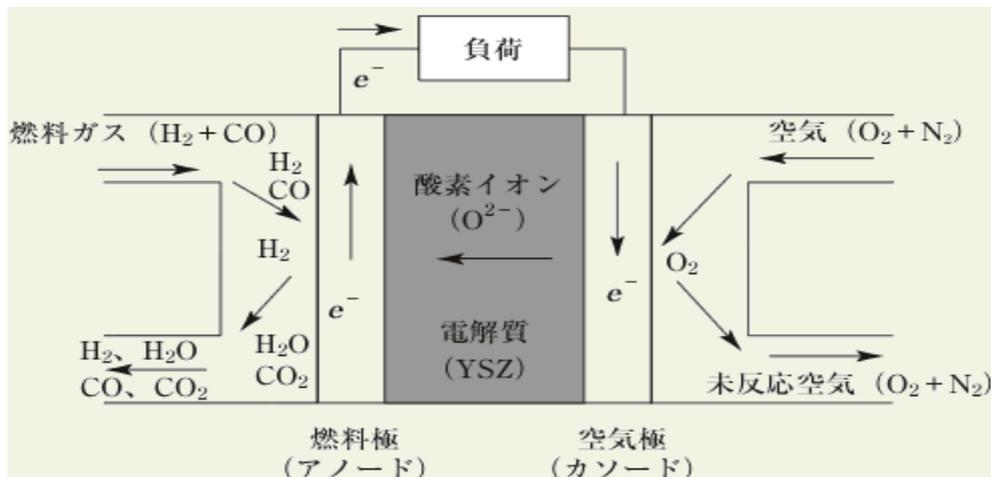
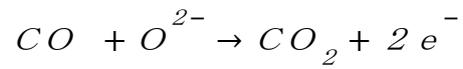
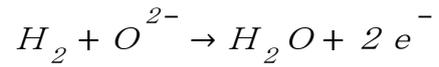
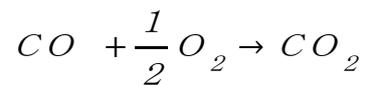
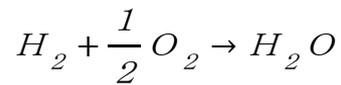


図 44 固体酸化物形燃料電池の動作原理 (出典：日本電気技術者協会)

空気極では酸素と電子から酸素イオンが生成されます。



電池全体の反応は次のようになります。



最終的に水（蒸気）と炭酸ガスが発生します。窒素の分離が不要なので、炭酸ガスは蒸気を凝縮させることで比較的容易に分離回収ができます。

3章3節 燃料電池車（FCV）

燃料電池自動車(FCV : Fuel Cell Vehicle)は、燃料電池で水素と酸素の化学反応によって発電した電気エネルギーを使って、モーターを回して走る自動車です。ガソリン内燃機関自動車が、ガソリンスタンドで燃料を補給するように、燃料電池自動車は水素ステーションで燃料となる水素を補給します。

3. 3. 1. 燃料電池車のしくみ

一般的な自動車は、ガソリンや軽油を燃料としてエンジンを動かして動きますが、燃料電池車では図のように、水素を燃料電池に与えて電力を生み出し、モーターを動かす仕組みです。

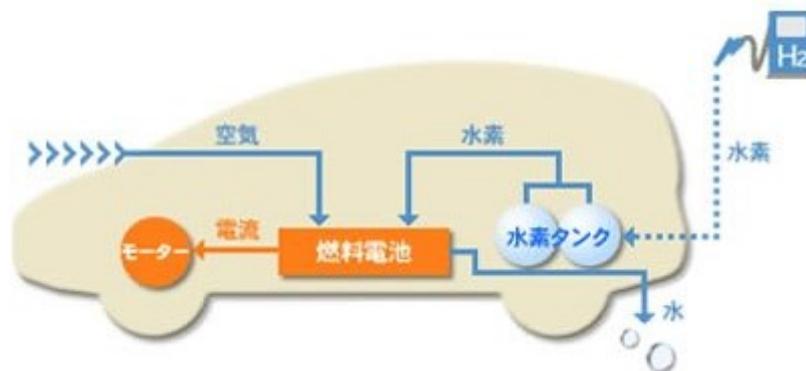


図 45 FCV のしくみ（出典：水素・燃料電池実証プロジェクト）

電気自動車（EV）も電力でモーターを駆動していますが、大きな違いは蓄電池を用いているという点です。燃料電池車（FCV）は燃料電池を用いて自ら発電を行います。電気自動車（EV）は蓄電池に蓄えておいた電力を使ってモーターを動かしています。

ハイブリッドカーはガソリン車と電気自動車の中に位置する車です。エンジンも蓄電池も搭載されているため、ガソリンを使って動かすこともできますし、電力を使って動かすこともできます。

3. 3. 2. 燃料電池車・電気自動車・ハイブリッドカーの違い

いずれの自動車も、ガソリン車やディーゼル車と比べて、はるかに高い環境性能を持っています。

	燃料電池車 FCV	電気自動車 EV	ハイブリッドカー HV
動力	モーターと燃料電池	モーターと蓄電池	モーターとエンジン
必要物	水素	電気	電気とガソリン
補給方法	水素ステーション	家庭の電源	ガソリンスタンド
航続距離	やや長め	短め	長め
環境性能	非常に良い	非常に良い	良い
車体重量	重い	重い	やや軽い
普及状況	開発中	市販済み	市販済み
主な車種	日産 X-TRAIL トヨタ FCHV-adv ホンダ FCXクラリティ	日産 リーフ 三菱 i-MiEV トヨタ コムス ホンダ EV Plus ダイハツ ハイゼット	日産 セレナ トヨタ プリウス ホンダ インサイト スズキ ランディ ダイハツ アルティス

図 46 各車の比較（出典：燃料電池. net）

3. 3. 3. 燃料電池車（Fuel Cell Vehicle）

燃料電池で発電した電力でモーターを回すことによって走行します。燃料電池は水素と酸素（空気）で発電を行うため、燃料電池車にはガソリン車にガソリンを補給するような形で水素を補給します。走行時に排出するものは水だけで、二酸化炭素や有害ガスを出さないことから、非常に環境に優しい自動車とされています。

3. 3. 4. 電気自動車（Electric Vehicle）

燃料電池車と同様に非常に環境に優しく、既に市販されているのが電気自動車です。電力でモーターを動かすという点は燃料電池車にも電気自動車にも共通していますが、水素ステーションが必要となる燃料電池車と異なり、自宅のコンセントから充電することができるというメリットもありますが、充電時間が長く走行可能距離が短いのが弱点です。

3. 3. 5. ハイブリッドカー（Hybrid Vehicle）

環境性能は燃料電池車や電気自動車に劣りますが、加速性や航続距離などといった走行性能に優れているのがハイブリッドカーです。ガソリン車と同様にガソリンでエンジンを動かしますが、同時に発電を行うことで蓄電池に電力を蓄え、その電力を使ってモーター

を回して走行するということも可能です。街中で気が付くと、まわりがみなハイブリッドカーということも珍しくありません。

4章 蓄電池

「電池」には大きく分けて「一次電池」と「二次電池」の2種類があります。一次電池は使い切りタイプの充電不可能な電池のことで、日常生活でよく用いられる乾電池やボタン電池などがあります。

一方、二次電池は充電して繰り返し利用可能な電池のこと。これが『蓄電池』で、充電式電池とも呼ばれます。蓄電池は、自動車のバッテリーやノートパソコン、携帯電話、自動車など幅広く使われています。使い続けるにつれて電池が少なくなるため、アダプターなどを使って充電することが特徴です。

4章1節 2つの蓄電池

蓄電池には鉛蓄電池やリチウムイオン電池、ニッカド電池、ニッケル水素電池など、様々な種類がありますが、スマートハウスに関連する住宅用の大容量蓄電池としては、主に鉛蓄電池とリチウムイオン電池が使われています。

	鉛蓄電池	リチウムイオン蓄電池
大きさ	13~17リットル/kWh	3~4リットル/kWh
重さ	25~30kg/kWh	4~10kg/kWh
寿命	短い	長い
価格	安価	高価
設置環境	屋内	屋内

図 47 鉛蓄電池とリチウムイオン蓄電池の特徴（出典：セキスイハイム）

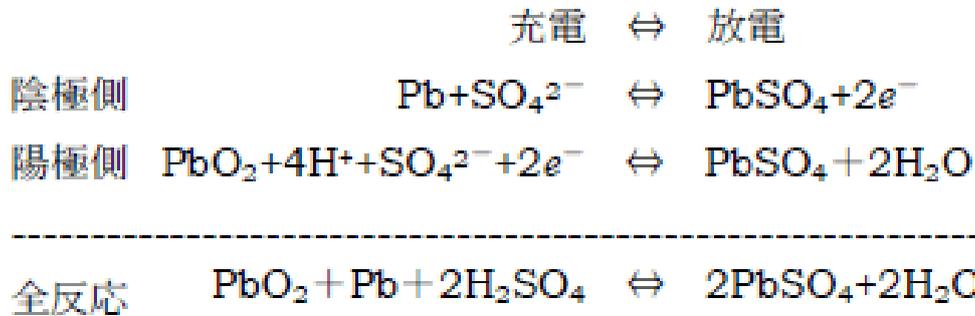
以下、2つの蓄電池について、まとめます。

4. 1. 1. 鉛蓄電池

1859年にフランスのプランテによって発明された鉛蓄電池は、現在一般に流通している蓄電池の中で最も古い歴史を持ち、自動車の普及が進んだ20世紀中盤以降に車載用の蓄電池として需要が拡大しました。現在でも世界市場では二次電池の中で最大の市場規模（国内では第2位）を持ち、民生用から産業用まで広い分野で中～大容量の蓄電デバイスとし

て広く利用されています。

◇原理 鉛蓄電池は希硫酸の水溶液（電解液）に浸した金属鉛の化学反応を利用した電池です。鉛蓄電池の化学反応式を示します。



この反応式で左側から右側に反応する現象が放電にあたり、右側から左側へ反応する現象が充電の反応式です。鉛蓄電池は放電が進むに従って電解液の硫酸の濃度が低下するので、液の比重が低下します。逆に充電が進むに従って電解液の硫酸の濃度が上昇するので、電解液の比重が上昇します。充電が完了すると両電極の硫酸鉛はほとんど分解されて元の活物質に戻ります。更に充電を続けると水の電気分解が生じて、陰極板から水素ガス、陽極板からは酸素ガスが発生します。このため、電解液の水成分が減少します。

◇構造 図に鉛蓄電池の構造を示します。

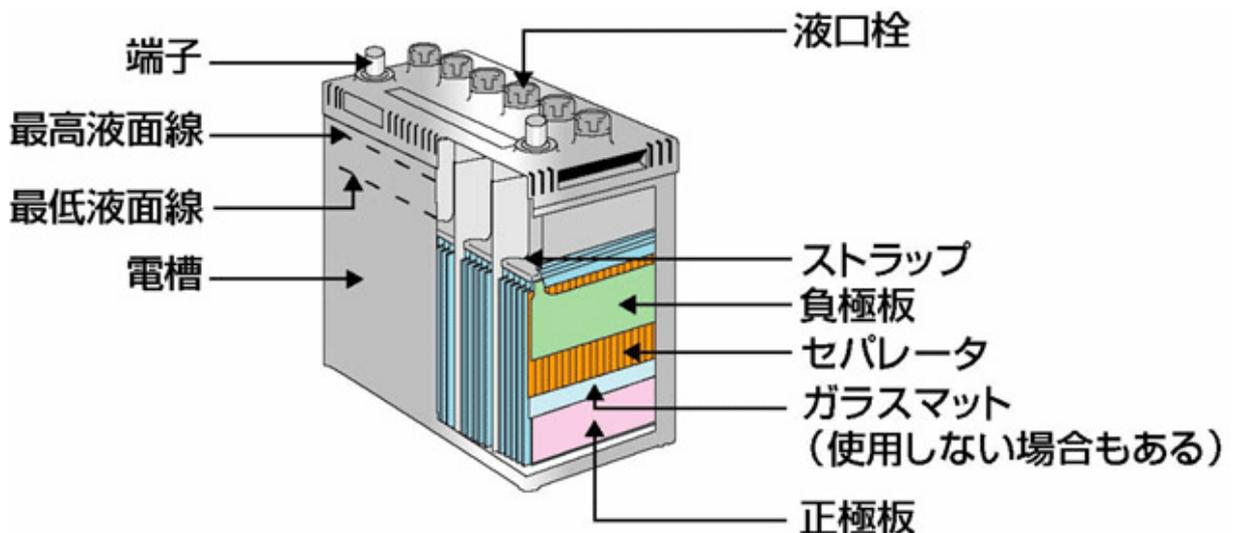


図 48 鉛蓄電池の構造（出典：電池工業会）

(a) 電極

鉛蓄電池にはペースト式とファイバクラッド式の 2 種類の電極が主に用いられています。

①ペースト式極板：鉛または鉛とアンチモンの合金、鉛とカルシウムの合金などに鉛酸化物を混ぜて希硫酸で練り上げ、ペースト状にしたものをグリッド（格子）に充填して乾燥させた後、化学合成したものです。ペースト式で作られた電極は鉛蓄電池の陰極に使用されますが、自動車用、小型運搬車用の蓄電池などは陽極板にも使用されることがあります。

②ファイバクラッド式極板：ガラス繊維またはプラスチック繊維で作った多孔性のチューブを鉛合金製櫛状格子（心金）に通し、このチューブと心金間に鉛粉を充填します。そして、このチューブを化学合成して活物質化することで電極として形成したものです。チューブに活物質を充填する構造であるため活物質の脱落がなく、活物質と電解液との接触性がよいのが特徴で、この極板は振動を受けることのない据置き形の鉛蓄電池の陽極のほか、振動を伴う車両などの鉛蓄電池にも用いられます。

(b) 隔離板（セパレータ）

陽極と陰極との間の短絡を防止するために設けられます。隔離板は長時間使用しても劣化せず、不純物の溶出がなく、また両極間のイオンの導通を妨げないものでなければなりません。このような性質を持つ隔離板の材料として、多孔性ゴム板、多孔性プラスチック、樹脂セパレータなどがあります。

(c) 電 槽

耐酸性がよく、機械的強度が強いポリエチレン、ポリプロピレンや繊維強化プラスチックが用いられます。

(d) 電解液

鉛蓄電池では純粋な希硫酸が電解液として用いられます。この電解液は比重で管理され、20℃における比重値を標準としています。完全充電時の電解液標準比重は1.280になっています。

◇特性

(a) 起電力

鉛蓄電池の公称起電力は2.0Vです。必要に応じて複数のセルを直列に接続して高電圧を得ています。車両用の12Vの鉛蓄電池は内部を仕切り板で仕切り、鉛蓄電池を6セル直列に接続した構造をとっています。

(b) 充電特性

鉛蓄電池の起電力を E [V]、充電電流を I [A]、鉛蓄電池の内部抵抗を R [Ω] とすれば、充電電圧 V_c [V] は、

$$V_c = E + IR \quad [\text{V}] \quad (\text{式 1})$$

で求めることができます。

鉛蓄電池の充電は、通常 5 時間から 10 時間程度をかけて行います。鉛蓄電池を一定の電流（定電流）で充電したときの鉛蓄電池の端子電圧と電解液比重の一例を図に示します。充電が進行して端子電圧が 2.35V を超えると鉛蓄電池からガスが発生するとともに、端子電圧が急上昇します。更に充電が進むと一定電圧、一定比重に落ち着きます。この状態になると電解液中の水の電気分解が起こり、鉛蓄電池の温度が急上昇します。

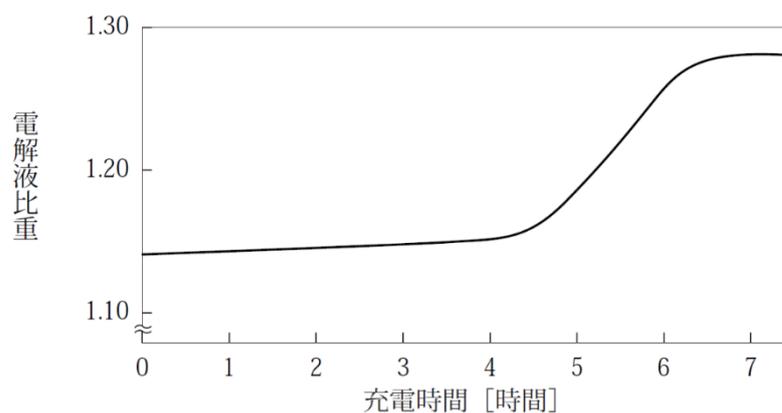


図 49 充電時間と電解液比重（出典：日本電気技術者協会）

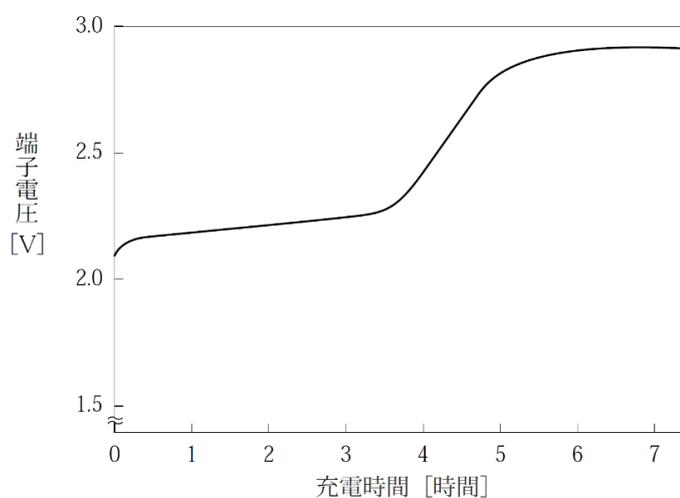


図 50 充電時間と端子電圧（出典：日本電気技術者協会）

図が示すように電解液の比重は、充電が進むにつれて上昇していくことが分かります。

(c) 放電特性

鉛蓄電池の起電力（開路電圧）を E [V]、放電電流を I [A]、鉛蓄電池の内部抵抗を R [Ω] とすれば、放電時の端子電圧 V_d [V] は、

$$V_d = E - IR \text{ [V]} \quad (\text{式 2})$$

で求めることができます。

鉛蓄電池 1 セル当たりの放電容量は、放電終止電圧を低くとれば大きくすることができますが、放電終止電圧の低下に伴って、蓄電池はより深い放電をすることになるので、蓄電池の活物質の劣化の進行が早まり寿命が短くなるおそれがあります。このため表に示すように放電率によって放電終止電圧を定めています。

表 4 放電率と放電停止電圧（出典：日本電気技術者協会）

放電率	放電終止電圧 [V/セル]
1 時間放電率	1.55
3 時間放電率	1.65
5 時間放電率	1.70

また、放電電流が大きくなると（式 2）に示すように電池の内部抵抗によって端子電圧が低下するので電池容量も低下します。図に放電率と放電容量を表した一例を示します。この図は 5 時間率（5HR）を基準電池容量（100%）としています。図に示すように短時間で放電させると鉛蓄電池の容量低下は著しくなります。

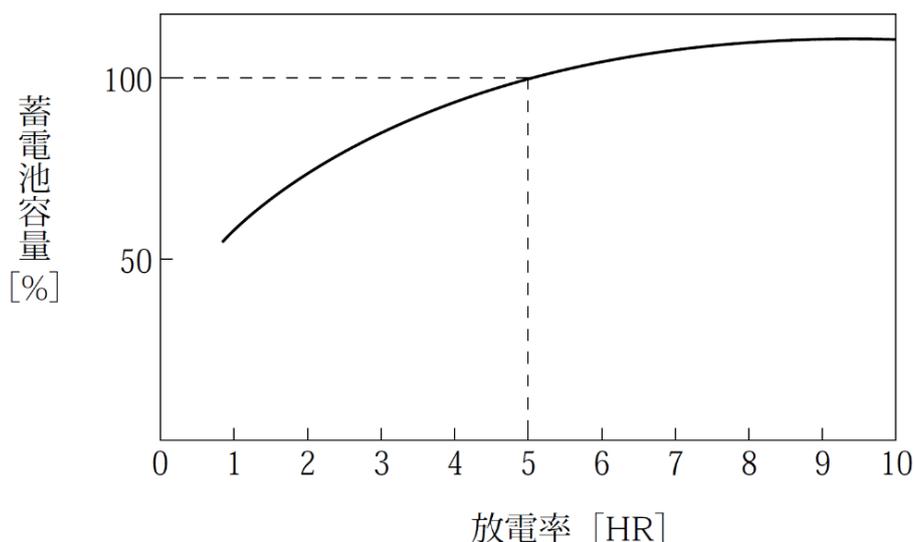


図 51 放電率と放電容量の関係（出典：日本電気技術者協会）

◇長所・短所

長所	<ol style="list-style-type: none"> 1. 他の二次電池と比較して容量当たりの価格が安く、コストパフォーマンスが良い。 2. 微小電流～大電流まで広い範囲で放電が安定している。 3. メモリ（※）効果が無い。 ※継ぎ足し充電したことで、放電中に、一時的な電圧降下を起こす現象のこと。
短所	<ol style="list-style-type: none"> 1. 他の二次電池と比較して体積エネルギー密度、重量エネルギー密度が小さい。 2. 過放電に弱く、過放電が発生すると性能が大きく低下し、回復しない。 3. 電解液に硫酸を使用するため破損時の危険が高い。 4. 極寒地では電解液の凍結・破損の危険性がある。

4. 1. 2. リチウムイオン蓄電池

リチウムイオン電池はニッケル水素電池が開発された翌年の1991年に、ソニーと旭化成により実用化され、携帯電話やノートパソコンなどのモバイル機器を中心に利用が拡大しました。質量・体積あたりの電力量をあらゆるエネルギー密度が大きく、軽量・小型にも関わらず、高い電圧を取り出すことができます。

現在一般に利用されている二次電池の中では最も歴史の浅い電池ですが、その普及は急速で、2010年の日本国内の金額ベースでの市場規模は二次電池全体の47%を占め第1位と

なっています。

◇原理 リチウムイオン二次電池は図に示すように正極（リチウム化合物）と負極（炭素化合物）との間をリチウムイオン（ Li^+ ）が電解液中を移動することで充放電を行います。つまり、充電時には正極のリチウムイオンが電解液中を移動して負極が電源から電子を受け取ります。一方、放電時は負極から電子を放出する（負荷に電流が流れる）と同時に電解液中をリチウムイオンが移動して正極に到達します。すると正極は元のリチウム化合物に戻ります。

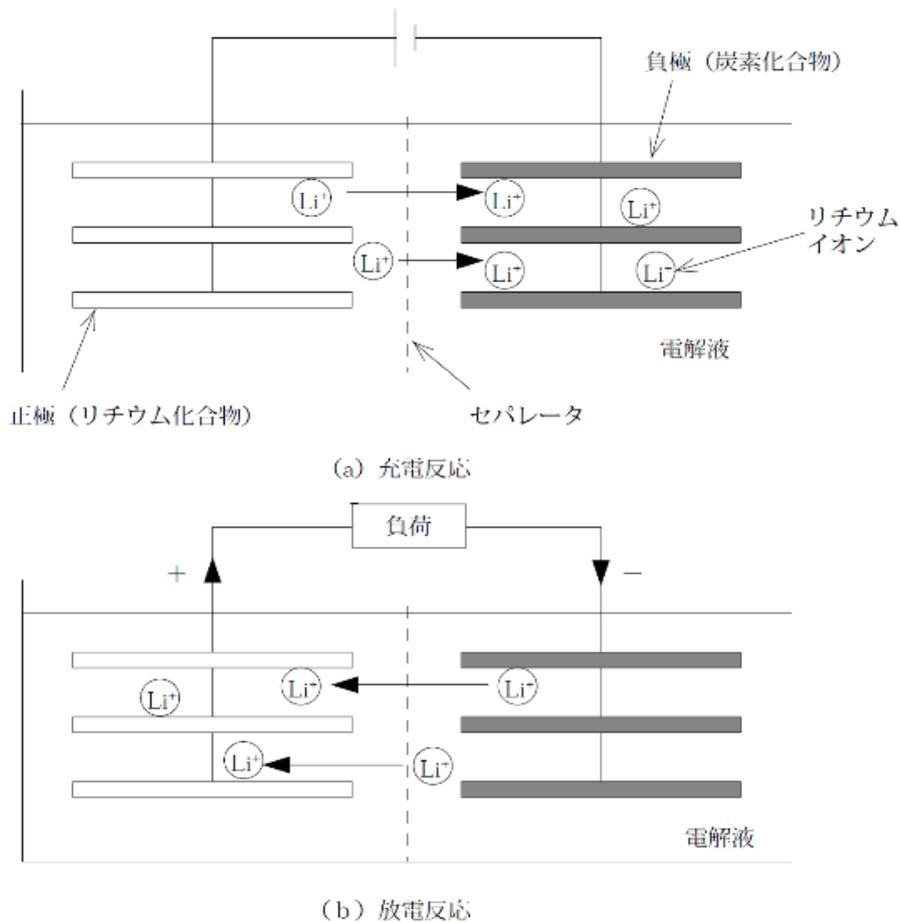
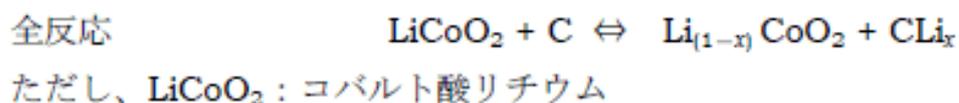
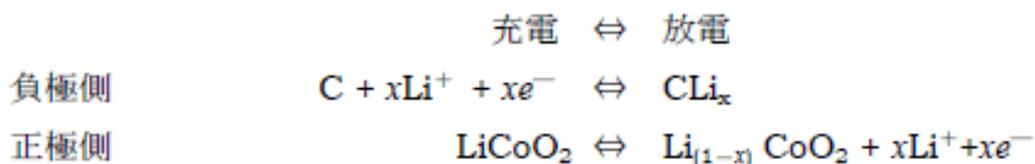


図 52 リチウムイオン蓄電池の反応（出典：日本電気技術者協会）

リチウムイオン蓄電池は、このように充放電のときにリチウムイオンが移動することで電子の受け渡しを行う方式をとっています。このためロッキングチェア形またはシャトルコック形と呼ばれます。これはイオンが揺り椅子やバトミントンのシャトルコックのように互い違いに移動することに由来しています。リチウムイオン蓄電池は充放電に際してイオンが正極と負極間を移動するだけなので、充放電の繰返し回数が多く取れ、高いサイクル特性を得ることができるわけです。リチウムイオン蓄電池放電反応式を以下に示します。



◇構造 リチウムイオン二次電池は正極にコバルト酸リチウムを、負極に特殊カーボンを薄くシート状にした電極を用います。そしてこの正極と負極との間に極薄いシート状のセパレータを挟み込み、これら3層から成る構造体を捲回してケースに収納した構造をとっています。また、この電池は電池単体の電圧が3.7Vと高いので、電解液に水酸化物質を用いると電気分解してしまうため、電解液には有機溶媒が用いられています。

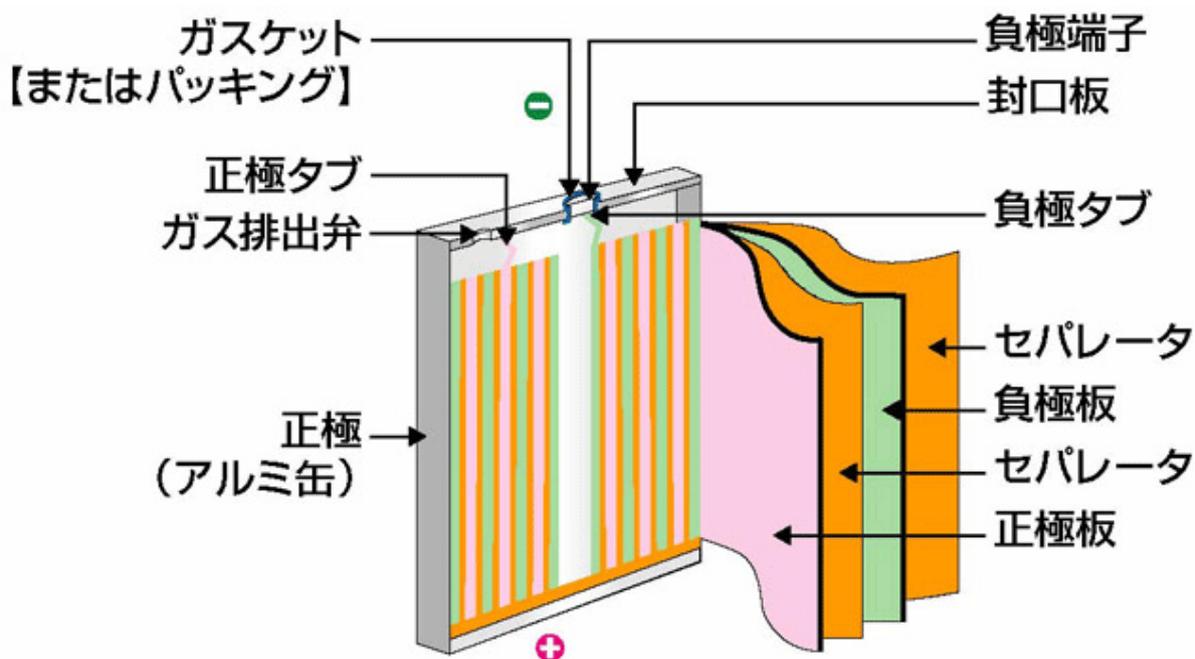


図 53 リチウムイオン蓄電池の構造 (出典：電池工業会)

◇特性

(a) 充電特性

リチウムイオン二次電池の充電方法は充電初期に定電流充電を行い、その後、一定の電圧に到達したところで定電圧充電に切り換えます。このような充電方法は電池の劣化を防止する目的で行われます。この電池の充電特性は図に示す

ように、定電流充電時に電池電圧が徐々に上昇して、所定の電圧に到達したところで定電圧充電に切り換える方式をとっているため定電圧充電になると充電電流が徐々に減少します。そして所定の充電電流値以下になったところで満充電としています。

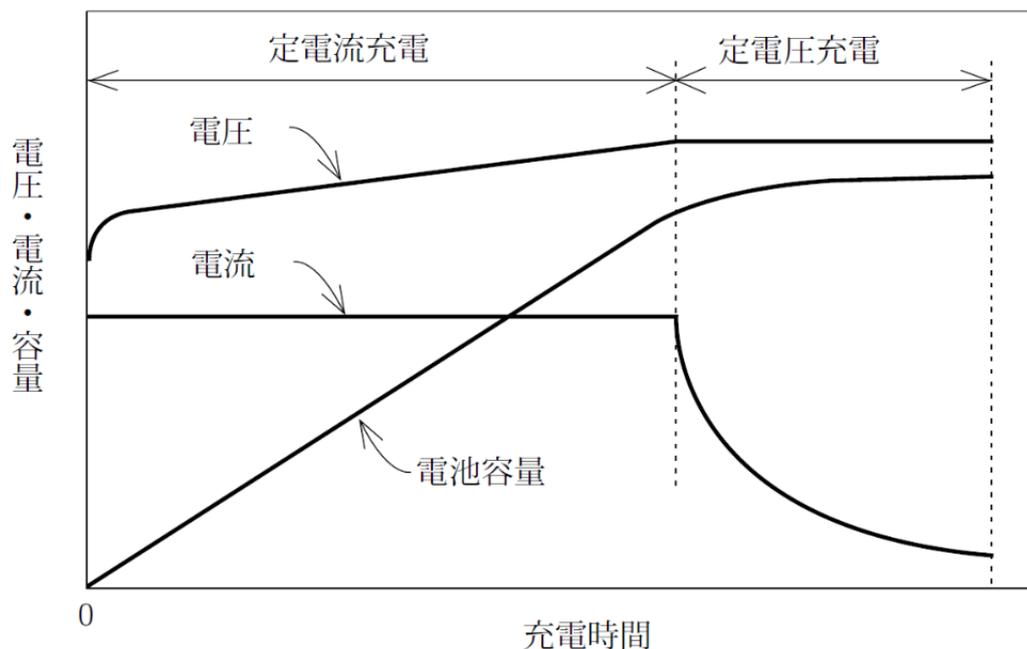


図 54 リチウムイオン蓄電池の充電特性（出典：日本電気技術者協会）

(b) 放電特性

リチウムイオン二次電池は放電当初の電圧低下がみられますが、その後ほぼ一定の電圧を維持します。そして放電終期になると電圧が急激に低下します。また、放電電流が大きくなるほど電池容量が低下します。

(c) 温度特性

電池の周囲温度が常温付近のとき電池容量が最も大きく、低温のとき電池容量が減少する特性があります。

(d) 寿命目安

メーカーが発表しているリチウムイオン電池（単セル）の寿命は、300 サイクルでおおむね 70～80%、500 サイクルでおおむね 50～70%となっています。

◇長所・短所

長所	<ol style="list-style-type: none">1. 他の二次電池よりもエネルギー密度が高い。2. メモリ効果が無い。(使用可能状態で繰り返し充電しても劣化しにくい。)3. 自己放電がニッカド電池やニッケル水素電池の 1/10 程度と非常に少ない。
短所	<ol style="list-style-type: none">1. 過放電、過充電共に異常発熱の危険性があり、保護回路が必須。2. 大電流放電に適さない。3. 低温特性がニッカド電池、ニッケル水素電池よりも劣る。

4章2節 V2H (Vehicle to Home)

Vehicle to Home とは、電気自動車 (EV) やプラグインハイブリッド自動車 (PHV)、燃料電池車 (FCV) などの自動車が蓄電池に蓄えた電力を家庭用電力として利用することを指します。

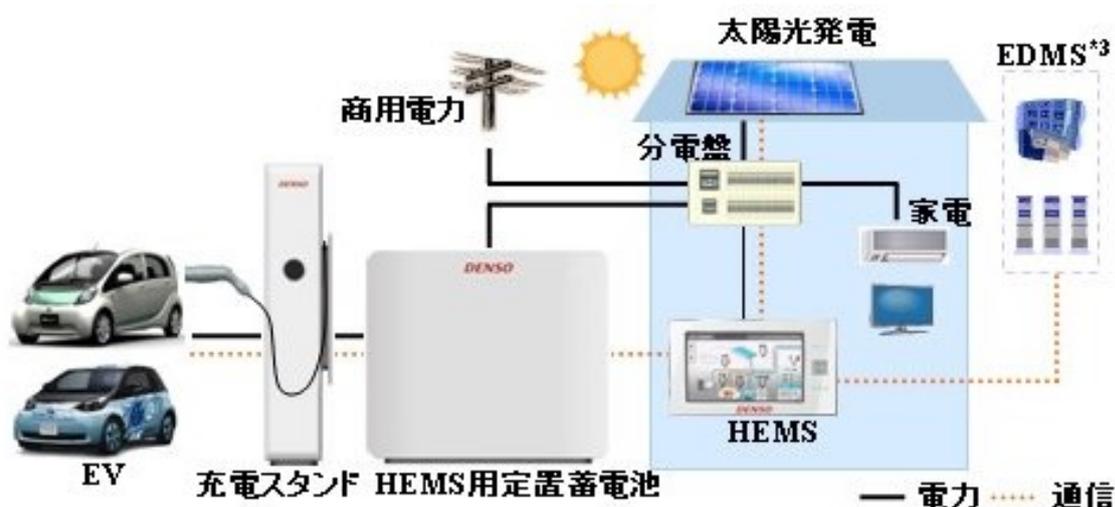


図 55 V2H 概念図 (出典: DENSO)

自動車を敷地内に駐車している間は、自動車の蓄電池を家庭内の発電システムの一部として、充電・放電するという双方向な電力のやりとりを行うことができます。

この V2H を実現するには、自動車の蓄電池が蓄積している直流の電力を家庭で利用できる交流の電力に変換しなくてはなりません。そのため双方向の直流/交流変換回路 (インバーター) を用意する必要があります。

また、次世代自動車には、V2H を含む 3 つの機能も期待されています。

表 5 次世代自動車に期待される 3 機能 (出典: 次世代自動車振興センター)

名称	電気の流れ	期待される機能
Vehicle to Home (V2H)	BEV・PHEVから家庭へ	車両の電力を家庭用の電力供給減として利用。電力系統への連携。 逆潮流の有無でその役割等に違いあり。
Grid to Vehicle(G2V)	グリッドからBEV・PHEVへ	車両への単純な充電だけではなく、太陽光・風力発電からの余剰電力のバッファ等として利用。
Vehicle to Grid(V2G)	BEV・PHEVからグリッドへ	車両から電力系統に電力を供給。電力系統の周波数調整・需給調整等へ利用。(アンシラリーサービス)

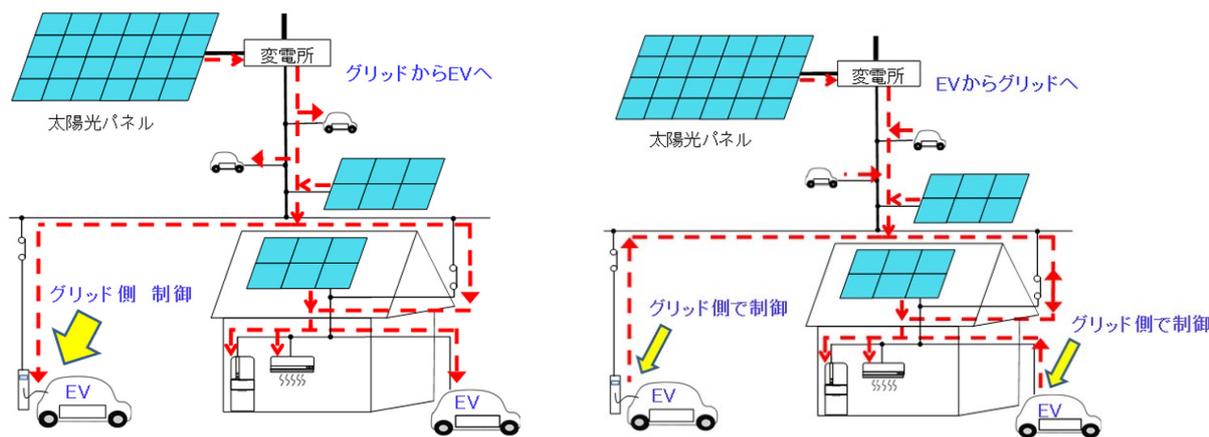


図 56 G2V (左) と V2G (右) (出典：次世代自動車振興センター)

4. 2. 1. スマートハウスでのV2H

家庭でV2Hを導入する大きなメリットは、安価な深夜電力で自動車を充電しておく、昼間のピーク時に電力会社からの電力を使わずに済ませるピークシフトが可能になることが挙げられます。

さらに、自動車と太陽光発電などの自家発電システムを接続し、電気供給力を上げることもできます。昼間は太陽光発電システムからの電力をメインに利用しながら自動車を充電し、夜間は自動車からの電力を家庭で利用するとよいでしょう。

また、停電時に蓄電池の電力がどうしても足りなくなった場合、充電ステーションまで自動車を走らせて充電してこることも可能です。非常時に電力を供給しやすい点は自動車ならではの強みです。

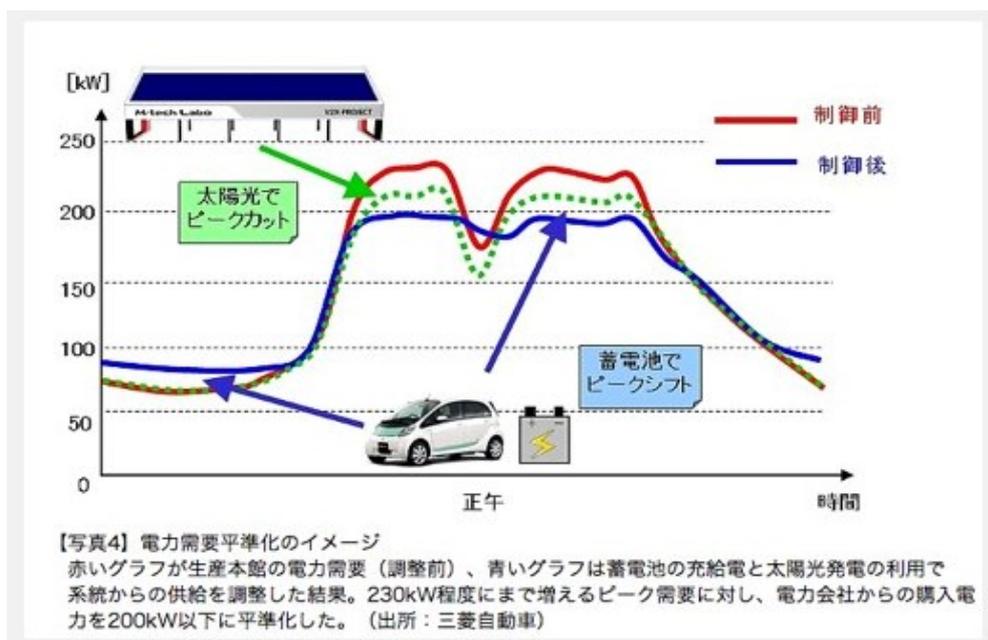


図 57 消費電力の平滑化 (出典：三菱自動車)

4. 2. 2. 自動車メーカーの取組

◇日産

ニチコン製「EVパワーステーション」を活用したV2Hシステム「LEAF to Home」の販売を開始。

環境	設置場所	本体:屋外 中継ボックス:屋内	
	周囲温度	-10℃～+40℃	
	周囲湿度	30～90%	
	騒音	約40dB	
機構	外形寸法・質量 (幅×高さ×奥行き)	本体 650×781×350mm 約60kg	
		中継ボックス 710×370×150mm 約10kg	
充電部	入力電圧	単相AC200V、50Hz/60Hz	
	最大出力電力	6kW	
家庭供給部	出力電圧	単相3線式 単相AC100V×2、50Hz/60Hz	
	最大出力電力	6kW(3kW×2)	

図 58 日産の取組 (出典: 次世代自動車振興センター)

◇三菱

i-MiEV、MINICAB-MiEV用のオプションとして外部への電力の出力が可能な『MiEV power BOX』を発売。



『MiEV power BOX』主要諸元

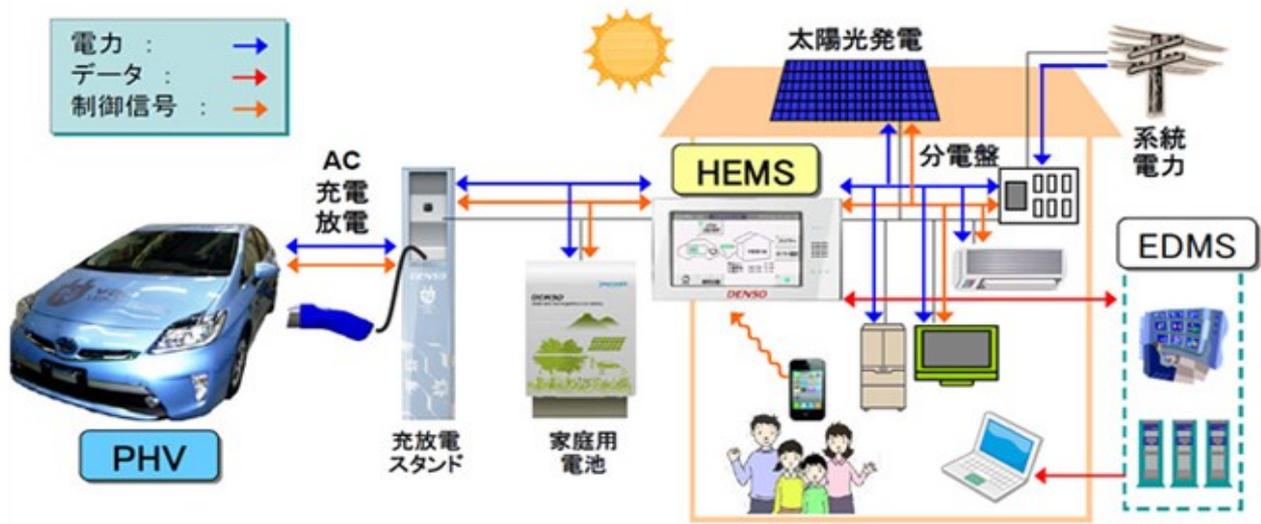
外形寸法	395mm×334mm×194mm
接続ケーブル長	1.7m
重量	11.5kg (本体9.5kg、ケーブル部2kg)
出力電圧	AC100V
最大出力	1500W (15A)
出力端子	AC100Vコンセント1個

(メーカー希望小売価格: 149,800円、消費税込)

図 59 三菱の取組 (出典: 次世代自動車振興センター)

◇トヨタ

PHVやEVなどと約10世帯の住宅との間で電力を相互供給する、『豊田市低炭素社会システム実証プロジェクト』を開始。



※PHVは満充電およびガソリン満タン状態で、1台で一般家庭の日常使用電力約10kWhの約4日分に相当する電力を供給することが可能

図 60 トヨタの取組 (出典：次世代自動車振興センター)

◇ホンダ

エネルギー需給を総合的にコントロール可能な「ホンダスマートホームシステム (HSHS)」を導入した実証実験住宅を埼玉県さいたま市に完成させ公開。



【可搬型インバータボックス】
 最大出力：9kW(連続7時間給電可能)
 給電系統：100V/30A×3系統
 コンセント：15A×6個、30A×3個
 重量：約50kg

※一般家庭のおよそ6日分の使用電力を供給できる、最大出力9kWの外部給電機能を装備した燃料電池電気自動車「FCXクラリティ」も発表。

図 61 ホンダの取組 (出典：次世代自動車振興センター)

※トヨタは2014年中に、ホンダは2015年度中に燃料電池自動車を販売開始すると発表しました。

5章 自己診断システム

長く使用していた PC が立ち上がらなくなった時、フロントパネルにある LED の点灯・点滅の仕方で、原因を表示してくれていたり、自己診断ソフトを実行することにより、不具合箇所を検出してくれたりします。また故障していなくても、例えばネットワークに接続できないとき、「診断と修復」などと書かれたメニューを選択すると、インターネットへの接続ができない原因を調べて、自動修復してくれたりもします。

PC は、コンピュータそのものですが、家電品や自動車などは組込系マイコンが多用されています。機能が豊富になる、使い勝手が良くなる、安全になる、省エネにつながるなどの理由で、多くの製品にマイコンが使われるようになると、不具合の理由も複雑になり、人が外部から調べただけでは、原因追及が難しい事が多くなりました。また、不具合ではなく使い方が適切でなかった場合、製品自体が事故を未然に防ぐ目的で動作を停止するなどということもあり、それを使う側が故障と勘違いしていることも多々あります。そのようなとき、誤使用の履歴データがマイコンのメモリに記録されていれば、その内容を確認してユーザーが安全に使えるように指導することもできます。複数の要因が重なった不具合発生なども、記録があれば、「1 箇所直して、まだうまく動かない。」などということがなくなり、安全性が向上します。

組み込まれているコンピュータが、各部分の正常動作をあらかじめ調べて、安全に動かすことができることが確認できると、全体の電源が入ったりするような機器は、身の回りに多くあるのではないのでしょうか。

この章では、自動車の自己診断機能について説明しています。

5章 1 節 自動車の自己診断機能

自動車の自己診断機能とは、自動車各部に取り付けられた ECU (Electrical Control Unit) にプログラミングされている機能のひとつです。英語で自動車の自己診断機能を表す用語：On-Board Diagnostics の頭文字を取って、OBD と略称されています。

例えばエンジンの燃料噴射システムを構成する各センサー及び各アクチュエータに、なんらかの異常が発生した場合、エンジンの ECU は異常の発生を記憶し、インスツルメンタルパネル（計器盤）の警告ランプを点灯させるなりして、ドライバーに異常の発生を知らせる。この警告灯は MIL (Malfunction Indication Lamp) と呼ばれています。

5章2節 ECU との通信規格

自動車の自己診断には、ECU との通信が不可欠です。そのための電氣的な接続規格やプロトコル、コネクタなどの物理的な規格はどのようにして、現在に至ったのでしょうか。

5. 2. 1. OBD1

1980年、ゼネラルモーターズ（GM）は、車両組立ライン上でのエンジン ECU の検査用のデータ通信システムを開発した。これは ALDL（Assembly Line Diagnostic Link）と呼ばれたが、あくまで品質管理のための機能で、アフターマーケットでの活用を意図したものではありませんでした。しかし、サービス用カプラーのある特定の端子同士を短絡することで、インスツルメンタルパネル（計器盤）の警告ランプの点滅回数で、故障コードを表示することができました。GM・キャデラックのマイコン制御エアコン仕様車では、エアコンのコントロールディスプレイで、各アクチュエータのテストを行ったり、各センサーのデータを表示したり、故障コードを表示することができました。

その後 ALDL のアップグレードを経て、1988年、自動車技術者協会（SAE）が標準化された診断コネクタと診断コードの標準化を勧告後、1991年カリフォルニア州大気資源局（CARB）が、1991年以降に州内で販売される新車に、エンジンについての自己診断機能の搭載を義務付けました。この時期の自己診断機能が『OBD1』と呼ばれ、診断コネクタ、診断コードも各社まちまちでした。

5. 2. 2. OBD2

こうした状況を改善するために制定されたのが OBD2（On Board Diagnosis second generation）ということになる。OBD2は各メーカーの枠を超えて、DTC（Data link coupler）と呼ばれる同じ形状・ピン配置の接続コネクタと、同じ故障コードを使い、故障発生時には同じように警告灯を点灯させる機能を実現させた。図のように、OBD2では次の5種類の通信規格が定められています。

規格	通信速度	最高電圧	メッセージ長	備考
SEAJ1850 (PWM)	41.6KBaud	5V	11bytes (CRC含む)	フォードの標準プロトコル、CSMA/NDA採用
SEAJ1850 (VPM)	10.4/41.6KBaud	7V	11bytes (CRC含む)	GMの標準プロトコル、CSMA/NDA採用
ISO 9141-2	10.4KBaud	バッテリー電圧	11bytes	クライスラー、日本車、ヨーロッパ車の標準プロトコル
ISO 14230	1.2~10.4KBaud	-	255bytes	ISO9121-2を改良した規格
ISO 15765 (CAN)	250/500Kbits	-	-	2008年から米国内で装備が義務付けられた。

図 62 OBD2 の通信規格（出典：Wikipedia）

アメリカのビッグスリー、GM、フォード、クライスラーがそれぞれ定めた規格、ヨーロッパの各メーカー及び日本の各メーカーが採用している規格、そしてCAN（Controller Area Network）です。これら5種類のうち、当初はいずれかの通信規格を備えていれば、その車種はOBD2に対応した自己診断機能を搭載していると認められていたため、各メーカーは、16ある端子の残りを、自社の専用スキャンツールで使う端子として割り当てています。

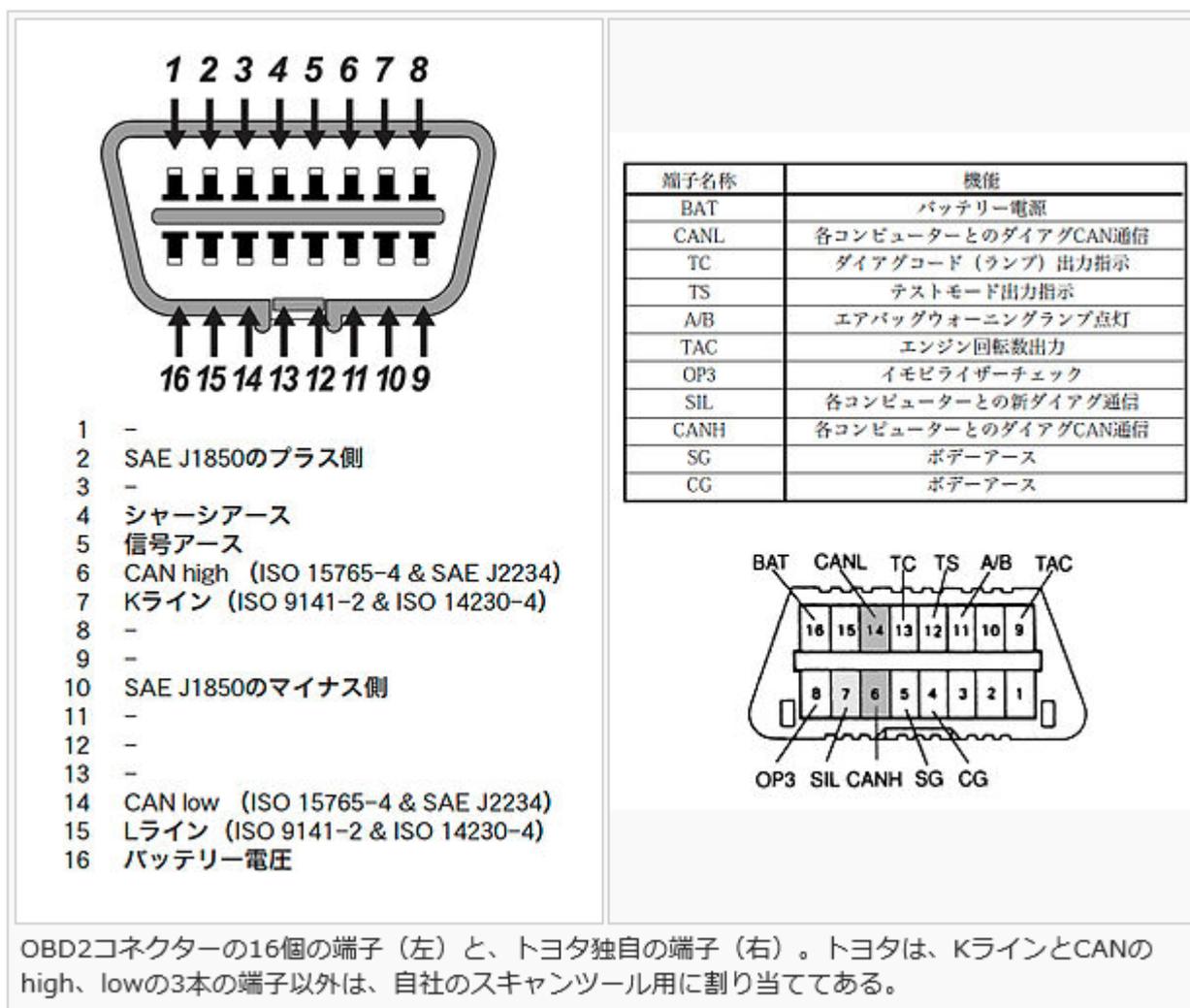


図 63 スキャンツールの端子割当（出典：Wikipwdia）

2008年、アメリカ合衆国が、国内で販売されるすべての乗用車、小型トラックのECU間のネットワークにCAN規格（Controller Area Network）の採用を義務付けたことにより、概ね2010年以降の新しいモデルは、CAN経由でスキャンツールとのデータのやり取りを行うようになっていきます。

5章3節 OBD2の故障コード体系

OBD2の故障コードは、DTC (Diagnostic Trouble Cord) と呼ばれ、アルファベット1文字と、4桁の数字から構成されています。4桁の数字を使えば、0000～9999まで実に一万個の故障コードを設定することができます。OBD1で使われていた2桁のコードとは違い、各メーカーは、非常に細かく故障コードを設定できます。

OBD2は元来、エンジンの排気ガスの有害成分抑制装置の不具合を検知するものであったので、当初はP (Power Train)で始まるDTC (通称Pコード) から設定されましたが、車の他の部位にまで拡大されました。現在DTCはPコードの他に、次の3種類のDTCが設定されています。

Cコード：Cで始まるコードで、シャーシ (Chassis) に関するDTC。ABSに関するDTCが該当します。

Bコード：Bで始まるコードで、ボディ (Body) に関するDTC。エアバッグに関するDTCが該当します。

Uコード：Uで始まるコードで、CANに関するDTC。

またアルファベットの次の数字が“0”の場合は、生産国に関係なく共通の意味を持つDTC (例：P0120：スロットルポジションセンサー回路異常) であり、これに対して“1”ないし“2”の場合は、各メーカーが独自に設定したDTCです。各メーカーは独自のDTCを設定した場合、SAEにそれを通知することが求められます。その情報はSAEから公開されることになっています。したがってもしスキャンツールでDTCの読み出しを行った場合、意味を把握できないDTCが読み出された場合、インターネットのDTCに関する検索サイトを用いて、その意味を把握することができるようになっています。

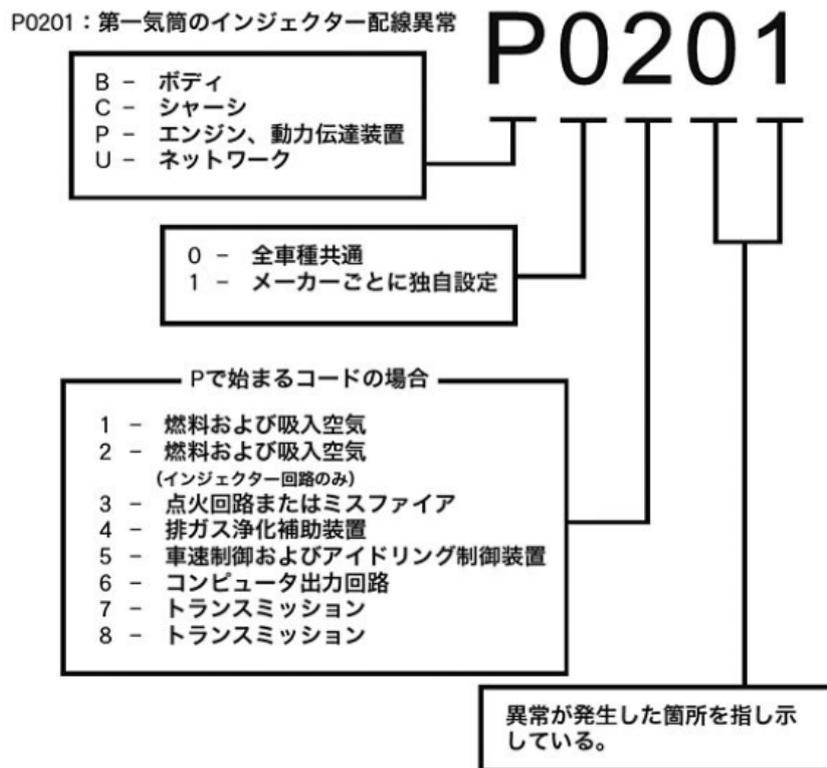
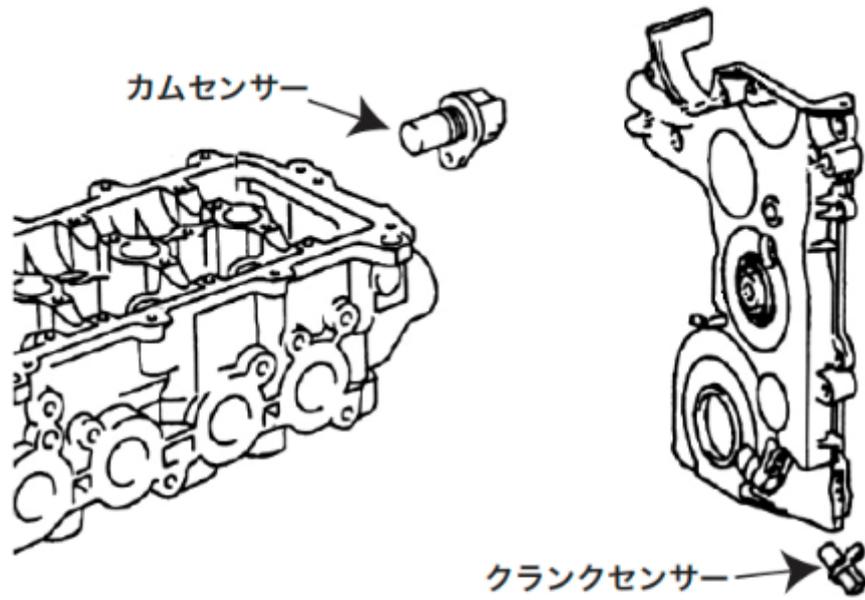


図 64 OBD2 の DTC の例 (出典 : Wikipedia)

OBD2 が搭載されることによって、自動車整備の現場では、確実かつスピーディーに行えるようになりました。例えばトヨタヴィッツ SCP10 系のエンジン関係の故障コード一覧表を見ると、この車は OBD2 スキャンツールを用いる方法の他に、“エンジンチェックランプの点滅回数を読みとる”という従来の OBD1 の方法でも、コードを読み取れるのがわかります。

その場合は従来からのトヨタ独自の二桁コードを読み取ることになりますが、“クランクセンサー異常”も“カムセンサー異常”も、同じ“12”が表示されてしまいます。つまり“12”が表示されたら、この二つのセンサーを両方とも点検しなければならないことになります。さて、それぞれのセンサーと取り付け位置は、一方はシリンダーヘッドの後ろ側、一方はクランクプリー付近と位置的にも離れていて、両方のセンサーの点検作業はかなり手間取ることになってしまいます。しかしスキャンツールを用いて OBD2 の DTC を読み取れば、その段階で点検箇所をどちらか一方に絞り込むことができます。



故障コード	診断内容
P0335 (12)	クランクセンサ回路異常
P0340 (12)	カムセンサー回路異常

↑ トヨタ独自の二桁コード

図 65 クランク角センサーとカムセンサーの位置関係 (出典 : Wikipedia)

5章4節 フリーズ・フレーム・データ

フリーズ・フレーム・データ (Freeze frame data) とは、OBD2 に対応したエンジン・コントロール・ユニットが記憶することができる情報です。各センサーあるいは各アクチュエータに異常が生じ、エンジン・コントロール・ユニットは DTC を記憶すると同時に、DTC を記憶した瞬間のエンジン冷却水温、エンジン回転数、負荷の状態 (高負荷時であったか、低負荷時であったか)、などの情報を記憶します。これがフリーズ・フレーム・データです。OBD2 に対応したエンジン・コントロール・ユニットは、必ずこの機能を持つことが求められていますが、記憶する情報の項目、項目数、表示方法は、各メーカーの意向に委ねられています。したがって実際の故障診断作業に当たっては、いつでも有益な情報をもたらしてくれるわけではありません。

5章5節 OBD スキャンツール

OBD2 の DTC を読取るには、ECU 内部のマイクロコンピュータと通信を行い、記憶された DTC を読み取り、それをディスプレイに表示する装置が必要です。この装置をスキャンツール (Scan Tools) と呼んでいます。



図 66 実際の OBD2 の DLC (データリンクコネクタ) とスキャンツール
(出典 : Wikipedia)

5. 5. 1. スキャンツールの種類

スキャンツールは、概ね次のように分類できます。

- ① 各自動車メーカーが、自社のディーラーのサービス工場向けに開発したもの。
機能的には最も充実していますが、一般には市販されません。また他社の車には使用することができません (例 : トヨタ TaScan)。
- ② 価格帯が 30~100 万円の汎用高級タイプ。
カラーディスプレイを備え、DTC の読み取りと消去、ライブデータ表示、アクティブテスト、さらにはオシロスコープ機能も備えている。ただし本体の他に、各メーカー毎のプログラムを別途購入する必要があり、それぞれのプログラムのバージョンアップ費用も発生します。
- ③ 価格帯が 10 万円以下の汎用ハンドヘルドタイプ。
テレビのリモコン程度のサイズで、小さなモノクロディスプレイが装備されています。主に DTC の読み取りと消去に特化したもので、いろいろなメーカーで生産されていて安価です。近年ライブデータ表示、アクティブテスト機能を備えたものが登場してきているそうです。
- ④ パソコンベースのタイプ。
日産のディーラーのサービス工場向け専用機 consult 3 もこのタイプである。DLC とパソコンの USB 端子とを結ぶコンバータとケーブル類、PC 用プログラム・インストールディスク、以上が一セットという形態で販売されています。



図 67 高級タイプスキャンツール（左）とハンドヘルド型汎用タイプ（右）
（出典：Wikipedia）

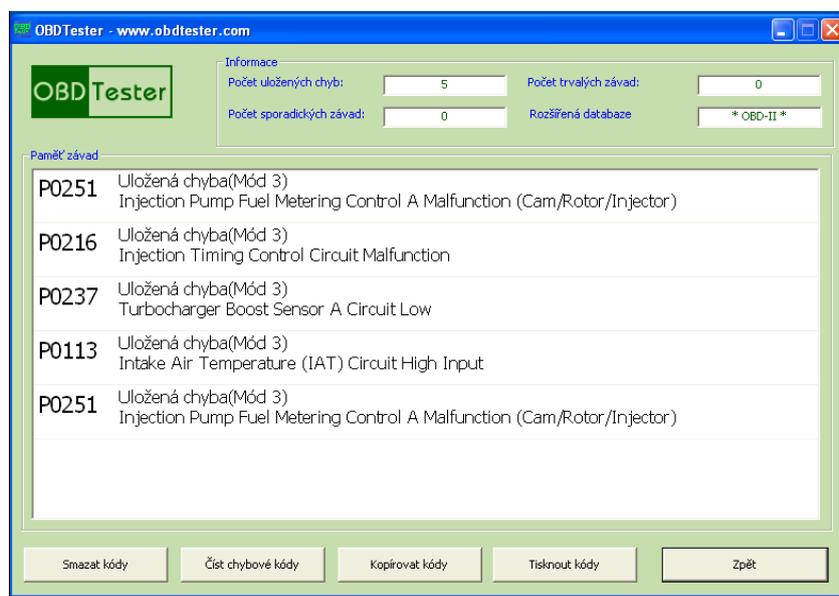


図 68 PC ベースのスキャンツール画面（出典：Wikipedia）

5. 5. 2. DTC の読み取りと消去

DLC には常時自動車側のバッテリー電圧がかかっているため、スキャンツールを接続すると自動的に電源が入ります。ECU が DTC を記憶している場合は DTC がディスプレイに表示されます。P コードの場合は、フリーズ・フレーム・データも確認します。表示された DTC にしたがって、ワイヤーハーネスやカプラーの修理が終わったら、ECU が記憶している DTC の消去を行います。その上で再びエンジンを始動し、必要に応じて試運転を行い、前回 DTC が記憶された運転状態を再現します。もし再び同じ DTC が記憶されるのなら、先ほどの修理作業が不完全ということになります。別の DTC が記憶されるのなら、それにしたがって修理作業を行います。そして DTC の消去を行い、試運転を行い、DTC が記憶されなければ、整備作業は完了です。

5. 5. 3. ライブデータ表示機能

ライブデータ (Live data) 表示機能とは、各センサーからエンジンの ECU に送られる信号をリアルタイムに表示する機能である。この機能を用いれば、例えば水温センサーが、始動時から完全暖機まで、どのようなデータを観測することが可能になる。何項目のライブデータが表示可能であるかは、それぞれの車のエンジンの ECU に依存するが、図の例の車では、合計 59 種類のライブデータが表示可能であるのが分かります (下図左やや右上の 25/59 という表示)。

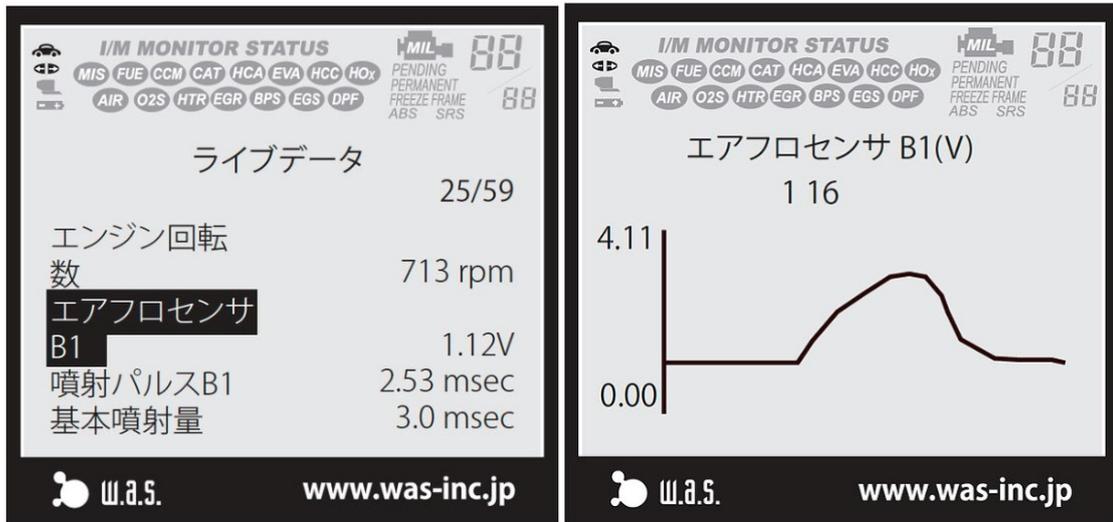


図 69 ライブデータ表示 (出典 : Wikipedia)

それぞれのライブデータは、グラフとして表示可能です。図はエアフローセンサーからの信号をグラフ表示させたものですが、電圧が低い状態はエンジンがアイドリング状態、電圧が山型に上昇し降下している部分は、アクセルペダルを踏み込んですぐ離れた時の電圧変化を示しています。

このように、エンジン回転数が上がって吸入空気量が増えた場合、エアフローセンサーの信号電圧が上昇するのが分かります。

6章 自動車と通信

現代の自動車の車載通信の主要規格である、CAN、LIN そして次世代自動車に必須の FlexRay が開発された背景、各々の通信方式、データフレームの内容までを解説します。また、4 節では家電品や様々な計測機器などに用いられるシリアル通信 (RS232c) を解説します。

6章1節 CAN

6. 1. 1. CAN の歴史

1983 年、CAN の開発はドイツの Bosch 社で始まりしました。2 年後の 1985 年に Full-CAN 規格が完成し、同年 Bosch 社と Intel 社による CAN 用マイコンの共同開発が開始されました。

この共同開発で Intel 社製のチップが 1987 年に完成し、翌年から Intel 社製 CAN チップ「82C526」の生産が始まりました。1989 年には Philips 社から Basic-CAN チップ「82C200」も登場しています。Full-CAN、Basic-CAN とは CAN コントローラの種類です。

量産車 (メルセデスベンツ S クラス) に CAN が搭載されたのは 1990 年になってからで、このときはエンジン、ギア (オートマチック)、エアコン回りでの採用でした。1994 年、正式に国際規格 (ISO 11898) となり、それ以降ヨーロッパ発の規格である CAN が広く自動車に採用されるようになりました。

6. 1. 2. ECU 間通信が求められた理由

CAN の開発が求められた主要因は、「ECU 間通信へのシフト」です。自動車の発展に伴い搭載される電装品の数が急増。これにより、高度な制御を行う要求が高まり、これらをコントロールする「電子制御ユニット (ECU: Electronic Control Unit)」が激増しました。これを背景に、各 ECU が単体で個々の電装品などを制御する方式から各 ECU 同士を専用線で結び、ECU 間のデータ転送による協調制御を行う方式 (ECU 間通信) へシフトすることとなりました。ECU 間通信が強く求められた具体的な理由は、次の 4 つです。

理由1：制御内容の高度化による入出力の増加とECUの大型化

自動車の高機能・高性能化に伴う制御内容の複雑化で、入力センサーや各種アクチュエータの数が増え、各 ECU はそれらの入出力の分だけコネクタのピンが必要となってきました。ピン数が増えればコネクタサイズが大きくなってしまいます。最終的にはコネクタの大きさによって ECU の規模が決められてしまうといった事態にまで陥ってしまいました。さらに、入出力用のデバイスも多数必要となるため、ECU 基板のサイズも大型化せざるを得ない状況になってしまいました。

理由2：複数のECUにまたがる制御の増加

従来は自動車の内部に搭載されている ECU が個別に制御を行う方法を取っていましたが、最近では 100 以上もの ECU が搭載されている車両も登場しており、各 ECU が個別に制御を行うことは困難になりました。

理由3：配線スペース、ECU収納スペースの制限

車両空間の快適性への要求が高まり、配線スペースや ECU の設置場所などが限られてきています。自動車の要求特性として安全・快適・環境が挙げられますが、特にコンパクトカーなどは「快適」要求が強く求められています。

理由4：故障診断・バックアップシステムへの対応

従来、故障情報を個々の ECU のメモリに記憶して、その情報から原因の究明を行ってきました。ECU の数が少ないうちは、1 つ 1 つ ECU の故障情報を確認して故障箇所を特定できたが、ECU の数が急激に増えたことで、個別の故障診断では故障箇所の断定に時間やコストが増加するようになりました。

6. 1. 3. 配線方式の違いによるメリット

各 ECU が単体で個々の電装品などを制御する方式から ECU 間通信へと変化してきましたが、ECU 間通信を実現する手段として初めに採用されていたのが、“必要な情報の数だけ ECU 間を配線で接続する「従来方式の配線」”でした。(下図)。

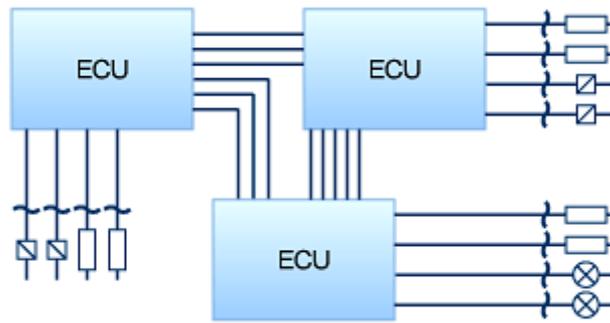


図 70 従来方式の配線（出典：ベクター）

この方式では、ECU やセンサーが増えるたびに配線が増え、配線コストの増加→車体重量の増加→配線スペースの確保問題へとつながってゆきました。さらに、配線と各 ECU やセンサーとの接続点も増え故障が起こりやすくなり「信頼性の低下」につながります。また、これだけ配線が入り組んだ状況では、改造や設計変更・拡張、故障診断時にも支障が出て、作業コストも増加します。

従来方式の配線によるデメリットをまとめると以下ようになります。

- ◆配線本数増加
- ◆費用増加
- ◆重量の増加、配線スペースの拡大
- ◆信頼性低下
- ◆設計変更・拡張の困難増加
- ◆故障診断の複雑性増大

こうして、従来方式の配線に代わる手段として考えられたのが「通信方式の配線」です（下図）。

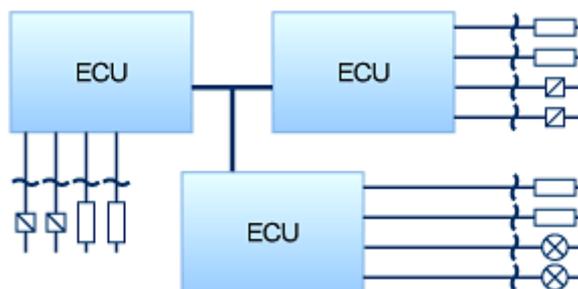


図 71 通信方式の配線（出典：ベクター）

通信方式の配線では、複数の部品間配線を 1 本の通信線にまとめ、各 ECU で通信線を共有している。このようにすることで配線本数が激減し、コストやスペース・重量の課題

を克服、さらには自動車の組み立てや設計作業の効率化、燃費の向上などが図れます。そのほかにも、ECU の小型化や設計自由度の向上、電氣的信頼性の向上、データ通信以外への用途拡大などのメリットも得られます。

以下に通信方式の配線によるメリットを示します。

- ◇柔軟性（拡張時や据え付け時）がある
- ◇ECU 同士の情報交換が可能
- ◇センサー信号の複数共有が可能
- ◇簡単な故障診断や統計処理が可能
- ◇送信エラーの検出が可能
- ◇エラー訂正が可能

この通信方式の配線、つまり ECU 間をネットワークで接続するための通信プロトコルとして考えられたのが CAN なのです。

図に主な車載ネットワーク概要を示します。

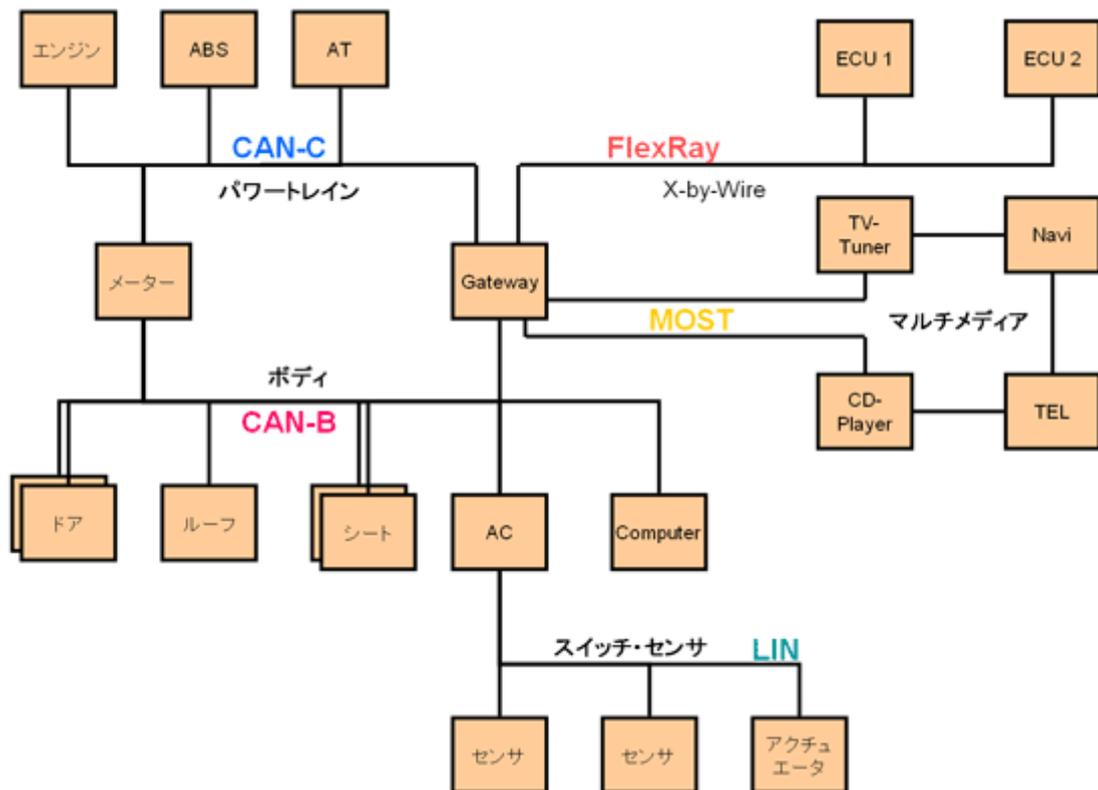


図 72 車載ネットワークの概要（出典：ベクター）

主にエンジンや ABS などの“パワートレイン系”で使用されているのが高速 CAN

「CAN-C (CAN-ClassC/SAE Class C)」。ドアやシートなどの“ボディー系”で使用されているのが低速 CAN「CAN-B (CAN-ClassB/SAE Class B)」です。

また、アクチュエータやセンサーなどの“スイッチ・センサ系”では低速・低コストの LIN が、ステアリング、ブレーキ、スロットルなどの“X-by-Wire”では高信頼・高速の FlexRay が、カーナビゲーションシステムやオーディオなどの“マルチメディア系”では MOST が使用されています。

X-by-Wire とは、機械による物理的な接続ではなく、通信により制御するシステムのことです。例えば、ハンドルとタイヤをステアリングシャフトでつないでタイヤを動かす代わりに、センサーで検知したハンドルの角度情報をステアリング制御用の ECU に伝えてモーターでタイヤの角度を調整することなどを行います。次世代自動車には不可欠の制御方式といわれています。

6. 1. 4. 各ネットワークの位置付け

車載ネットワークにもいくつか種類が存在しますが、それぞれの特長を通信速度とコストから比べてみましょう（下図）。

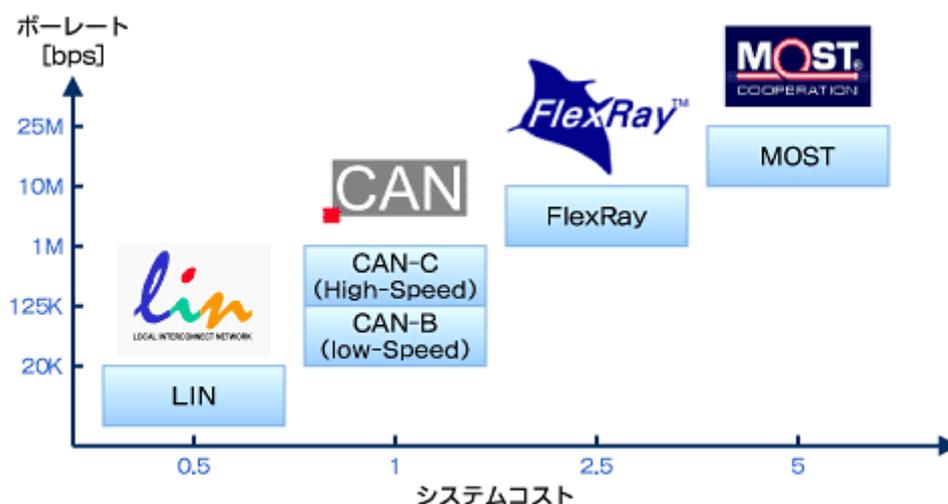


図5 通信速度とコストの比較
※ベクター・ジャパンの資料を基に作成

図 73 通信速度とコストの比較（出典：monoist）

一般的に、CAN のシステムコストを“1”とすると、LIN はその約半分。より高速・高信頼の FlexRay では CAN の約 2.5 倍、マルチメディア系の制御を行う MOST になると CAN の約 5 倍ものシステムコストが掛かります。

次にプロトコルごとの最大通信速度と CAN における通信速度比較を示します。

表 6 通信プロトコルの最大通信速度 (出典 : monoist)

通信プロトコル名	最大通信速度
CAN-B	125kbps
CAN-C	1Mbps
LIN	20kbps
FlexRay	10Mbps
MOST	24.8Mbps (50Mbps、150Mbpsも規格化)

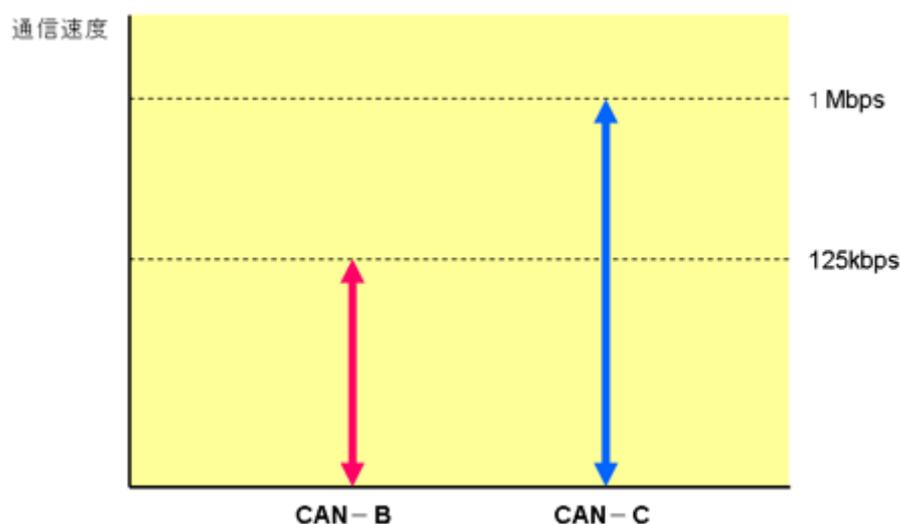


図 74 CAN-B と CAN-C の通信速度比較 (出典 : monoist)

このように、車載ネットワークは複数種類が存在し、それぞれ適応分野が異なっています。そして、各通信プロトコルがその特性やコストなどによって、適材適所で使われています。

以下、CAN の特徴について解説します。

6. 1. 5. ライン型構造

車載ネットワークに限らず、一般的にネットワークに接続される通信機器のことを“ノード”とといいます。車載ネットワークでいうと「電子制御ユニット (ECU : Electronic Control Unit)」がそれに当たります。そして、複数のノードを接続してネットワークを実現しますが、その方式にもいくつかの種類があって、例えばリング型、スター型、ライ

ン型などです。CAN では「ライン型」が採用されています。(下図)

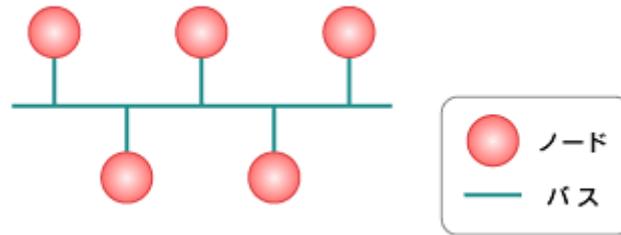


図 75 ライン型構造 (出典 : monoist)

ライン型によるネットワーク方式は、通信線 (バス : bus) に各ノードを接続していくことによりネットワークを構成できるため、ネットワークがシンプルで、その設計が容易に行えるというメリットがあります。車載ネットワークには、もともと“複雑な配線の解消”という目的があり、その点からしてもライン型構造は目的に合致しています。

6. 1. 6. マルチマスター方式

CAN ではライン型構造で接続される各ノードに平等なバスアクセスが可能な「マルチマスター方式」を採用しています。

マルチマスター方式を使用するメリットとしては、

- ◇各ノードが均一仕様で設計できる
- ◇各ノードに優劣がないため、イベント指向通信に向く
- ◇ノードの追加接続が容易

などが挙げられます。

つまり、開発時には各ノードを同一仕様で設計できることに加え、データ伝達を必要な場合に送信することができ、各ノードに優劣がないため自由度の高いネットワーク構成が可能となります。車載ネットワークでは、さまざまな使われ方や設計変更などのケースが考えられるため、このような自由度の高いネットワーク構成は非常に重要です。

6. 1. 7. CSMA/CA

通常、複数のノードから自由にデータが送信されてしまうとデータが衝突してしまうため、“通信線が使用中であればデータ送信をさせない”仕組みが使用されていますが、複数のノードから同時に通信線にデータが送信された場合には、これを防ぐことができません。

これに対して、CAN は「CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance)」を採用し、複数のノードから同時に通信線へデータが送信されても、その中

の“優先順位が高いものを（衝突させることなく）送信する”ようになっているので、複数ノードからの同時送信が起きた場合も、重要なデータに高い優先順位を設定しておけば、確実にデータ伝達が行われます。（下図）

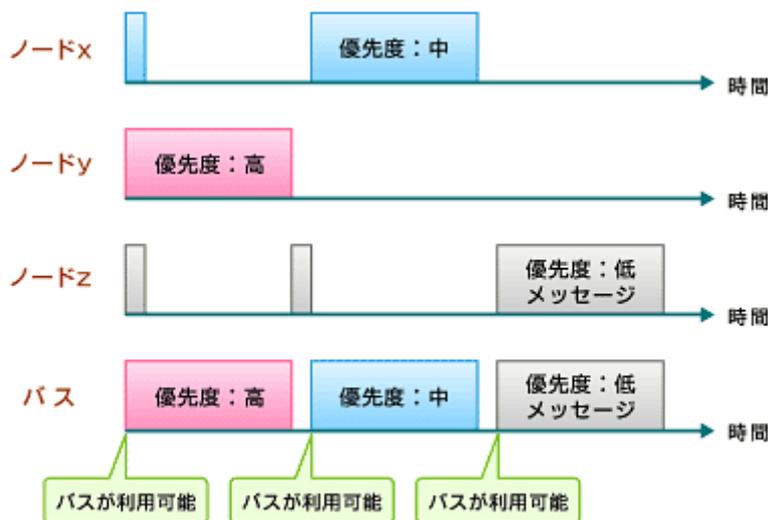


図 76 CSMA/CA（出典：monoist）

例えば、エンジン ECU からのデータが重要であれば、あらかじめ高優先順位を設定しておきます。そうすれば、仮にエンジン ECU からのデータとエアコン ECU からのデータが同時に送信されても、エンジン ECU からのデータは破壊されることなく通信線に送信することができます。

6. 1. 8. ID を使用したメッセージ・アドレッシング

ID を使用したメッセージ・アドレッシングとは、データの中に“ID（識別子）を付加して送信する”方式で、各ノード間でデータをやりとりする際に使用されます。データを受信する側は、この ID により「どのようなデータなのか?」「自分が使用するデータなのか?」を判断できます。

この方式を使用することで 1 つの送信ノードからのデータを複数の受信ノードが同時に受信できます。車載ネットワークで考えると、エンジン ECU からのデータを、メーター ECU とエアコン ECU が同時に受信して、それを使用することができます。こうして複数の ECU が協調して同時に制御を行うことができますようになります。

6. 1. 9. 耐ノイズ性に優れた物理層

パワートレイン系に使用される「High Speed CAN (CAN-C)」では、「2線式差動電圧方式」により通信を行っています。2線式差動電圧方式とは、2本線（ツイストペア）の各線に流れる電圧の差が“ある”か“ない”かによってデータを送信する方式です。この方式を使用することにより、外部からノイズが混入した場合でも、各線に混入するノイズの電圧はほぼ同一となる（各線の差電圧は変化しない）ので、ノイズによる影響を受けにくくなります。

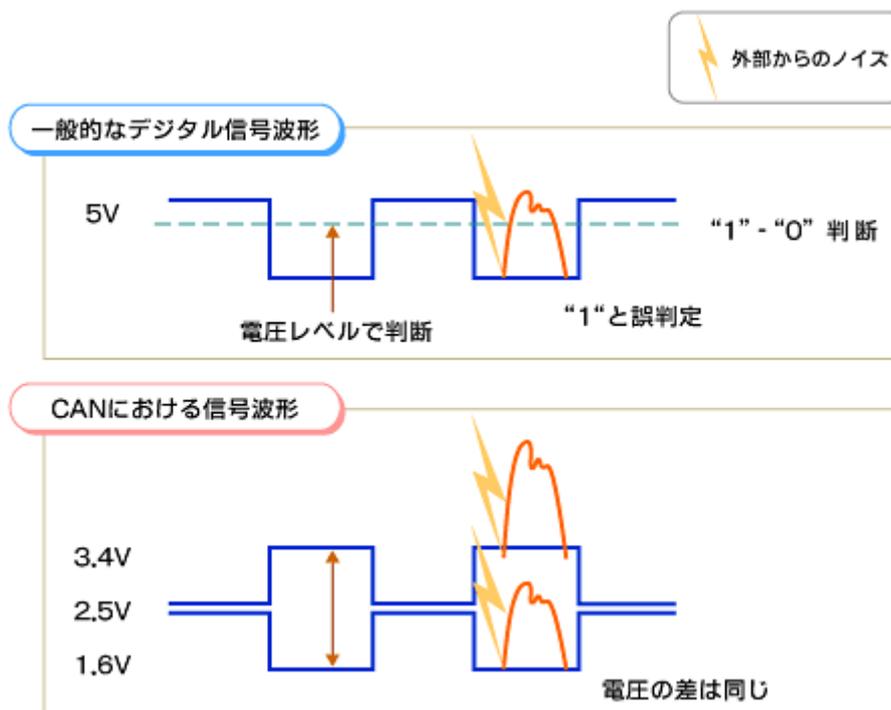


図 77 CAN における信号（出典：monoist）

車載機器では外部ノイズによる影響は避けて通れませんが、CAN ではそれらの影響を受けにくくする方式を採用しているため、信頼性の高いネットワークを構築することができます。

6. 1. 10. エラー検出メカニズム

安全面を重視する車載ネットワークでは、「各ノードで使用されるデータが何らかの異常によって間違っただま届き、正常な制御が行えなくなる」といった状態に陥ることは確実に避けなければいけません。

この対策として、CAN ではさまざまなエラー検出メカニズムを実装しており、これにより、ほぼ 100%に近い確率で各種エラーを検出することが可能となっています。例えば、

送信ノード自身が、送信するデータと通信線上に流れたデータが合っているかどうかを確認し、違いがある場合にはエラー検出を行うことができます。

6. 1. 11. データの一貫性

車両制御において、あるタイミングで各ノードが協調し、同時に制御を行う場合があります。

例えば、現在のエンジン回転数を使用して複数のノードが制御を行う場合、エンジン ECU から各ノードにエンジン回転数のデータが送信されます。通常、すべての受信側で正常に受信できれば問題ありませんが、もし1台のノードがエラーとなってデータを受信できなかつたらどうなるでしょうか。

「受信に失敗したノードのみエンジン ECU からデータを再送信してもらえばいい。」と思われませんが、これではエンジン回転数の変化が起きた後のデータを受け取る可能性があります。各ノードで使用するエンジン回転数データが異なってしまうこととなります。

これに対して CAN は、“もし1台のノードが受信に失敗したら、データを受信できた全ノードはそのデータを破棄し、全ノードが受信に成功するまでこれを繰り返す” ようになっていて、データの一貫性を保っています。

6. 1. 12. 通信速度など

◇1回に送信できるデータ量が最大 8bytes

◇最大通信速度 1Mbps

6. 1. 13. CAN の基本

[1] ドミナントとリセッシブ

CAN では、データの伝達にデジタル方式が使われています。送信されるデータは“0”と“1”の2進数に変換されて、通信線に送信されます。CANにおいて、この送信される2進数データの“0”のことを「ドミナント」、 “1”のことを「リセッシブ」と呼んでいます。

ドミナントは“優性”、リセッシブは“劣性”の意味で、ドミナントとリセッシブが別のノードから同時に送信された場合には、文字どおり“ドミナントが優先”される仕組みになっています。このルールを利用して、CANではいろいろな仕組みが実装されています。

[2] シグナルコーディング

CAN では「NRZ (Non-Return-to-Zero) 方式」により、送信したいデータを変換／送信しています。この方式は比較的シンプルに変換が行えるため、さまざまな通信プロトコルで使用されています。

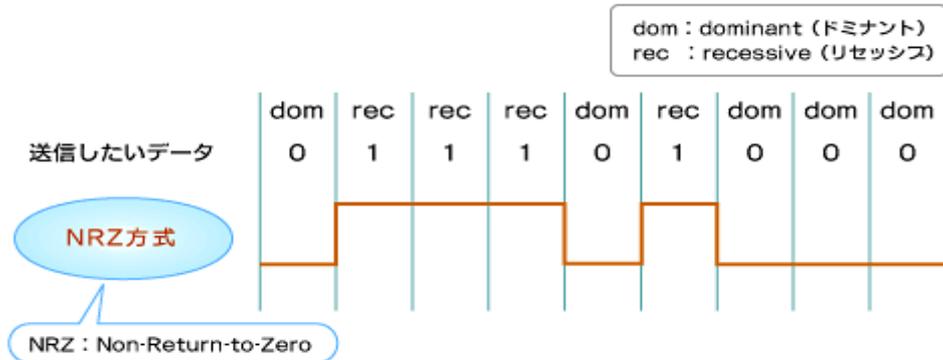


図 78 NRZ 方式 (出典 : monist)

NRZ 方式では、例えば送信するデータが“0001100”の場合、連続する“0”の部分は“0”のまま、連続する“1”の部分は“1”のままとなり、ドミナント状態やリセッシブ状態が連続することになります。状態が連続することによるデメリットもありますが、これを解消するための仕組みが用意されているのです。

[3] 通信速度

通常、通信速度は“bps (Bit Per Second)”で表されます。これは1秒間に何ビットのデータが送信可能かを表します。この数値が高ければ高いほど、より短い時間に大量のデータを伝えることができます。例えば、2bps の通信速度の場合は1秒間に2ビットのデータ伝達が可能となります。

下図のようにノード A が 1bps、ノード B が 2bps で動作する環境だと、ノード A から“1”を1個送信した場合、ノード B では“1”が2個受信されてしまいます。つまり、ノード A-B 間のデータ伝達は正常に行えません。通信を正常に行ううえでは、各ノード間の“通信速度を同一に保つ”ことが重要です。

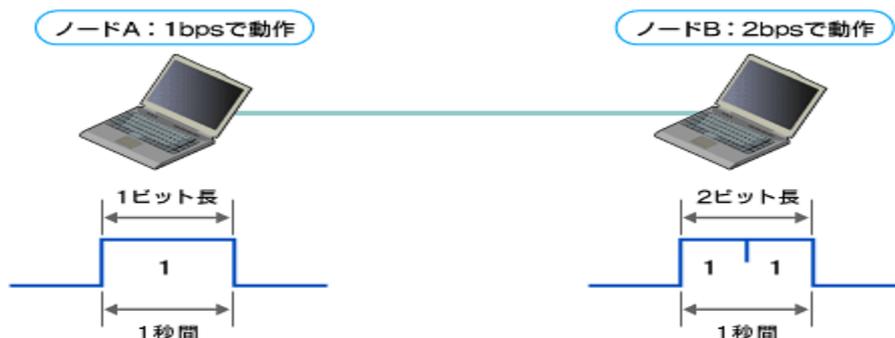


図 79 通信速度 (出典 : monoist)

[4] 同期

車載ネットワークでは、多種多様なノードが接続されています。そして、それぞれのノードは内部でプログラム処理などを行うための基準時間（以下、システムクロック）を作り出す“水晶発振子”を持っています。

この水晶発振子は比較的正確に時間を作り出すことができますが、車載ネットワークの各ノードはさまざまな場所に設置されているため、外気温などの影響を受けやすく、電源投入時からの時間変化などにより結果として各ノードのシステムクロックに違いが生じてしまいます。

CAN ではシステムクロックを使い、ドミナントやリセッシブの長さを1ビット分として構成していますが、もし、システムクロックが各ノードで違ってしまうと、それぞれの1ビットの長さが違ってしまうことになります。また、1ビットの長さによって前述の通信速度に違いが生じるため、正常なデータのやりとりができなくなってしまいます。

これを防ぐ制御が「同期」と呼ばれるもので、各ノード間のシステムクロックの違いを補正しています。CAN では“信号がリセッシブからドミナントへ変化するとき（“1” - “0” 変化時）に同期”を行っています。

[5] ビットスタッフィングルール

前述した NRZ では“0000”のように“0”が連続した場合、同じ状態が連続することになります。同期は“1” - “0”の変化にて行われるので、もし“0”が長時間連続した場合、同期ができない状態になってしまいます。同期ができないと各ノードの通信速度にずれが生じ、正常に通信ができなくなってしまいます。

そこで CAN では、「ビットスタッフィングルール」を採用しています。

これは、通信線上で同じ状態が5回連続した場合、それまで送信されていた状態と反対の状態のビット（“スタッフビット”）を1個挿入する仕組みです。図に実際の動きを示します。

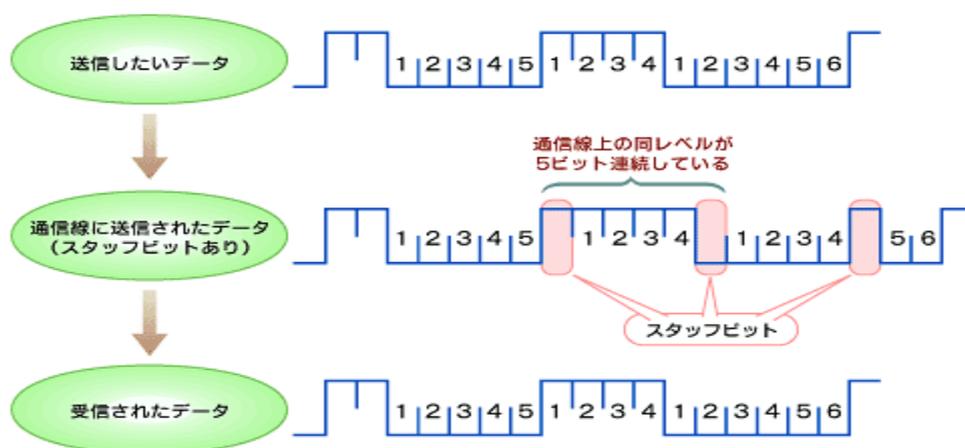


図 80 ビットスタッフィング (出典：ベクター)

例えば、“000000111111”と連続する場合は“00000101111101”と実際に送信されることとなります。(太文字は挿入されたスタッフビット)

また、“00000111111”と連続した場合は“0000011111101”となります。

これは、挿入されたスタッフビットも通信線上の状態としてカウントするからです。このように、ビットスタッフィングルールを採用することで、通信線上で同じ状態が延々と連続することを防止しています。

「ちょっと待てよ、これでは実際に送信するはずのデータがスタッフビットによって通信線上で変化してしまっているではないか！」と思うかもしれませんが。しかし、送受信側ともに同じルールで動作しているので、受信側は同じ状態が5回連続したら、次に受信されるビットはスタッフビットであると容易に判断できて、使用するデータにおいてはスタッフビットを除いたものを利用できます。

6. 1. 14. データフレーム標準フォーマット

CAN 通信において、データを送信する際の転送フォーマットをデータフレームと言います。そして、このデータフレームの形式には“標準フォーマット”と“拡張フォーマット”の細部で異なる2種類のフォーマットが存在しています。

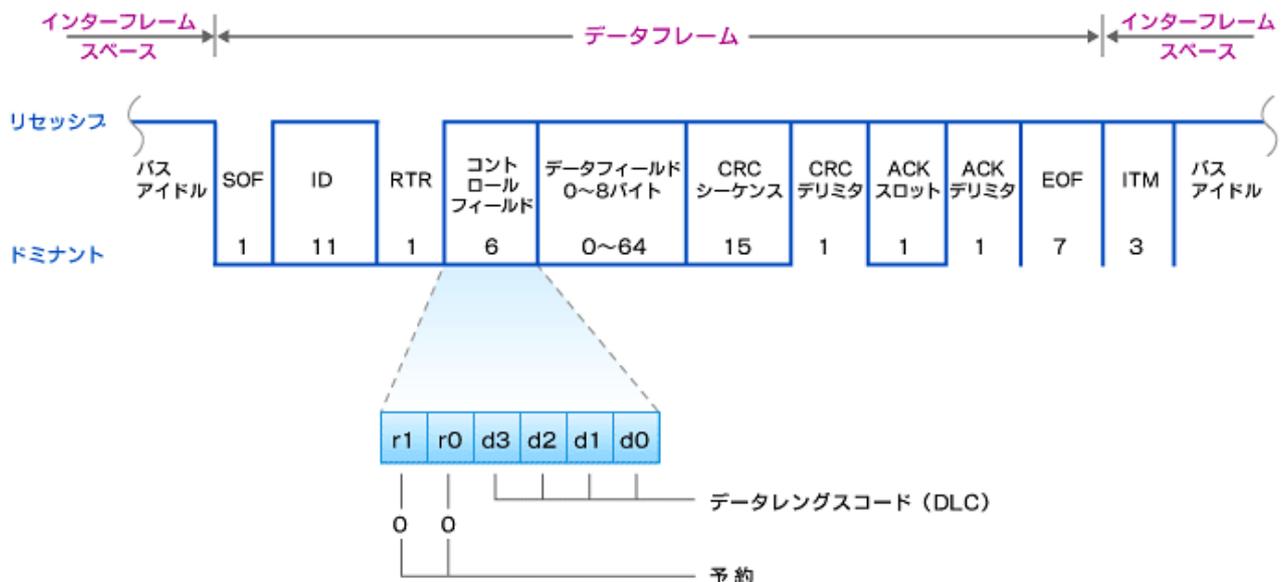


図 81 標準フォーマット (出典 : monoist)

図に描かれている上の線は“リセッシブ”を、下の線は“ドミナント”をそれぞれ表示しており、ドミナントにしか線がない部分はドミナント固定、リセッシブにしか線がない部

分はリセッシブ固定となっています。両方に線があるものは送信されるデータによってドミナントまたはリセッシブに変化することを示します。

各部の数値は、何ビット分使用するかの長さ（ビット長）を表しています。また、CANでは通信が行われていない場合、バスはリセッシブとなっており、これを“バスアイドル”と呼んでいます。

ノードからデータフレームが送信される時、最初に送信される部分は“開始”を表すためにドミナント状態とします。この部分を「SOF (Start Of Frame)」と呼びます。SOFがバスアイドルのリセッシブからドミナントへ変化することにより、受信側ノードは同期を行うことができます。

SOFに続き送信されるのは、「ID (識別子 : Identifier)」です。このIDはデータ内容や送信ノードを識別するために使用されるが、別の働きとして通信調停の優先順位を決定しています。標準フォーマットでは11ビット長でIDを構成していますが、後述する拡張フォーマットでは、このID (ベースID) と拡張IDの18ビット長でIDを構成しています。11ビット長の場合、IDの範囲は0x0~0x7FFとなり、2048種類の識別が可能です。

IDの後は「RTR」です。これは“Remote Transmission Request”の略で、データフレームとリモートフレームを識別するために使用されます。データフレームの場合には、RTRはドミナントとなっています。なお、RTRもIDと同様に通信調停に使用されます。

RTRに続くのは「コントロールフィールド」で、このコントロールフィールドは各1ビット長の予約ビットr1、r0と4ビット長の「DLC (Data Length Code)」から構成されます。標準フォーマットの場合、予約ビットr1、r0はともにドミナント固定になっています。

DLCは、コントロールフィールドに続く「データフィールド」において、何バイトのデータが送信されるかを表している。DLCの設定範囲は0~8となっており、データフィールドでは1バイト単位で0~8バイト(0~64ビット)のデータを送信できます。データフィールドは、送信されるデータの部分で、前述のようにDLCによって設定されたデータ長になります。また、データフィールド内では全バイトは最上位ビット(MSB)より送信されます。データフィールドは1バイトごとに長さを設定できますが、どのような形式でデータを割り当てるかについては設計者が決定できるようになっています。例えば、1バイトのデータを割り当てる場合では、そのまま1バイトとして使用、4ビット分をそれぞれ1ビットずつ使用、残り4ビットをまとめて使用、8ビットをそれぞれ1ビットずつ使

用などが可能です。

データフィールドの後は、「CRC (Cyclic Redundancy Check) シーケンス」が送信されます。CRC シーケンスは 15 ビット長であり、送信ノードが SOF、ID、コントロールフィールド、データフィールドの送信値より演算して、CRC シーケンスで演算結果を送信します。そして、受信ノードが送信ノードと同様に SOF、ID、コントロールフィールド、データフィールドの受信値から演算して、その結果を比較することで正常に受信できたかどうかの判断を行います。

CRC シーケンスの後には「CRC デリミタ」が送信されます。これは CRC シーケンスの終了を表す区切り記号で、1 ビット長のリセッピ固定となっています。なお、CRC シーケンスと CRC デリミタを併せて「CRC フィールド」と呼びます。

続いて、「ACK (Acknowledgement) スロット」が送信されます。この ACK スロットは 1 ビット長で送信ノードはこの部分でリセッピの送信を行います。ただし、受信ノードが CRC フィールド部分まで正常に受信できた場合は、ACK スロットのタイミングでドミナントを確認応答として送信することになっています。CAN ではドミナントとリセッピが同時に送信された場合はドミナントが優先され、ドミナントとなります。このことより、正常に通信を行っている CAN ネットワークでは ACK スロットはドミナントとなっています。しかし、この ACK スロットは 1 ビット長しかもたないため、CAN ネットワーク上の受信ノードがすべて正常に受信を行えたかの判断には使用できません。あくまでも、送信ノードから送信されたデータフレーム (CRC フィールドまでの部分) を正常に受信できたノードが存在することしか分かりません。

ACK スロット後には「ACK デリミタ」が送信されます。これは ACK スロットの終了を表す区切り記号で、1 ビット長のリセッピ固定となっています。CRC フィールドと同様に、ACK スロットと ACK デリミタを併せて「ACK フィールド」と言います。

データフレームの終わりには「EOF (End Of Frame)」が送信されます。EOF は 7 ビット長のリセッピ固定となっています。すでに説明した“ビットスタッフィングルール”は、SOF~CRC シーケンスの終わりまでの範囲で適用することとなっているため、バスアイドル中や EOF は適用外となります。

データフレームは SOF~EOF の範囲ですが、図 81 では EOF 終了後に「ITM (Intermission)」という表記があります。この ITM はデータフレームには含まれません。ITM は 3 ビット長のリセッピ固定となっており、この ITM 終了後にバスアイドル状態

6. 1. 16. リモートフレーム

リモートフレームはデータフレームの要求に使用され、基本構造はデータフレームからデータフィールドを除いたもの（もしくは、データフレームで DLC を 0、データフィールドが 0 バイトのもの）と同一になっています（下図）。

リモートフレームの ID は、要求するデータフレームの ID を設定し、リモートフレームの DLC は要求するデータフレームの DLC を設定します。データフレームと違う部分は、RTR がリセッシブ固定となることです。この違いにより、RTR を使用しデータフレームとリモートフレームの識別を行うことが可能となっています。



図 83 リモートフレーム（出典：monoist）

6. 1. 17. データフレームとリモートフレーム

CAN 通信における通信方法の基本は、データを必要としているノードからリモートフレームを送信し、それに対して該当するノードからデータフレームを返すという形になっています。リモートフレームはデータの要求、データフレームは要求に対してのデータの返信を行います。

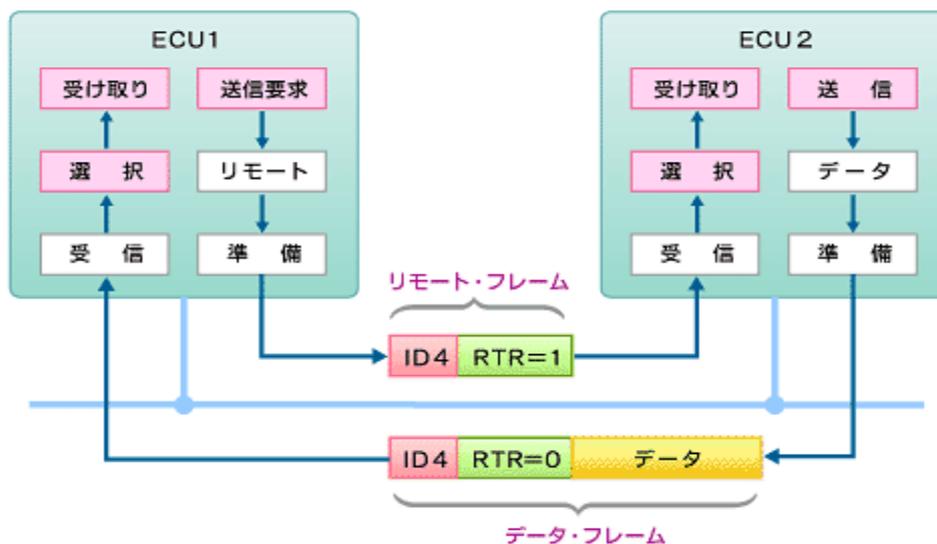


図 84 データフレームとリモートフレーム（出典：monoist）

この方式を使用すると、ノード内マイコンのリソースを送受信に占有されず、必要とな
 ときのみリソースを使用できて、バス占有率も下がることとなります。しかし、頻繁なデー
 タのやりとりにおいては、必要なデータフレーム数と同じリモートフレームが必要となる
 ため、かえってバス占有率が上がってしまう可能性もあります。

このため、制御の高度化により各ノード間の情報交換が非常に多くなっていくに従って、
 従来方式と違った方法で通信を行うことが考え出されました。

「従来方式と違う方法」とは

例えば、各ノードから自由にデータフレームを送信し、データフレームを必要とするノ
 ードが自由に受信する方式であれば、リモートフレームを使用せず、データフレームのみ
 で各ノードで情報交換が実現可能となります。

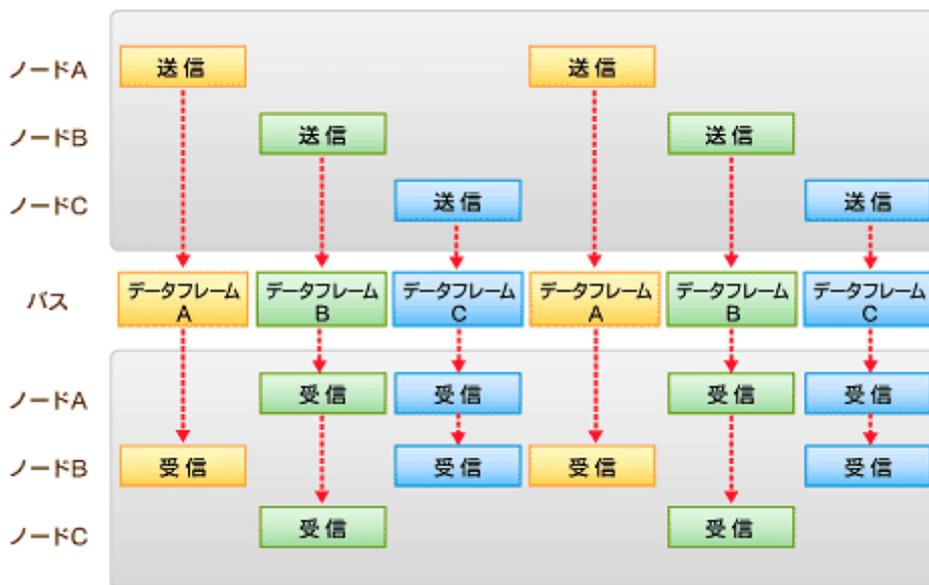


図 85 データフレームのみの送信 (出典：monoist)

リモートフレームとデータフレームによるやりとりを【注文してから調理され品物が出
 てくるレストラン】に例えたとすると、各ノードがデータフレームを送信して必要なとき
 にノードがデータフレームを受信する方法は、【回転寿司】のようなものといえます。いわ
 ば、各ノードはベルトコンベアに載せられて運ばれてきたデータフレームを必要なときに
 受信していることとなります。

6. 1. 18. 通信調停

CAN で採用している CSMA/CA では、バス使用中に他ノードはデータフレームやリモートフレームを送信することができない仕様になっています。しかし、実際には複数ノードから同時に送信されてしまうことを防ぐことができません。そのため、通信調停を行う必要があります。

CAN において通信調停に使用されるのは次の図の ID と RTR です。（ここでは標準フォーマットを例に解説します。）

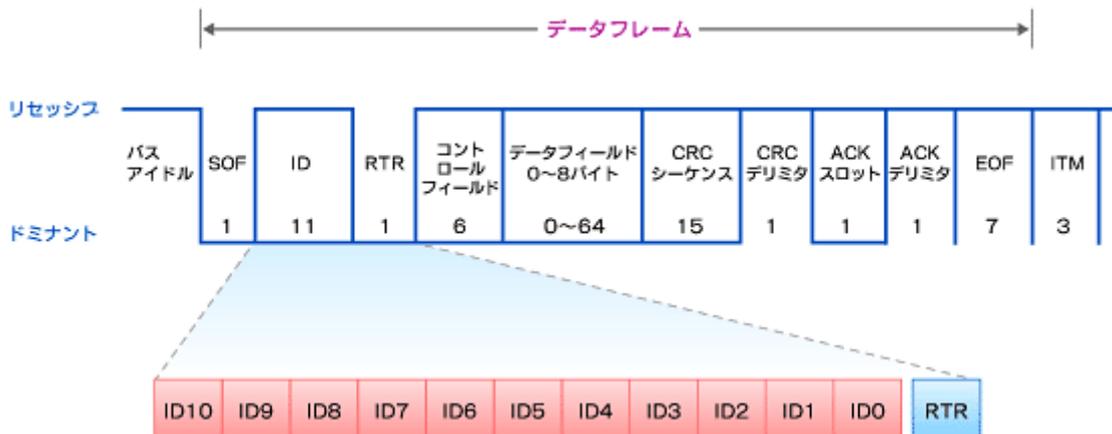


図 86 調停通信（出典：monoist）

実際の通信調停がどのように行われているのかを見てみましょう。

図に、ID 0x653 と 0x65B の 2 台のノードより同時にデータフレームが送信された場合を示します。

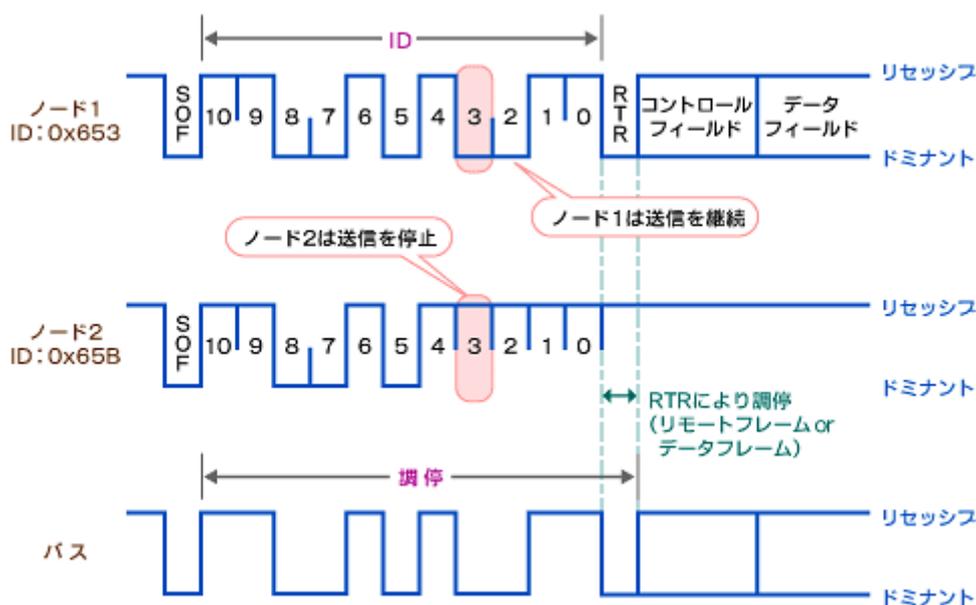


図 87 調停通信例（出典：monoist）

同時にデータフレームが送信される場合、複数ノードからの開始位置は同一になります。はじめに **SOF** が送信されますが、**SOF** は 1 ビット長のドミナント固定となっており、バスの状態はドミナントとなります。

各ノードは自身が送信したものとバス状態をモニタし比較しますが、**SOF** では各ノード自身が送信した内容のままになっていると判断し送信を継続します。

続いて、**ID** が 1 ビットずつ送信されますが、送信中の複数ノードの送信ビットが同一、例えば複数ノードからリセッシブが送信されれば、バスはリセッシブとなって送信した内容そのままとなります。

リセッシブとドミナントが別々のノードより同時に送信された場合、ドミナントが優先され、バスの状態はドミナントとなります。このとき、リセッシブを送信したノードは自身が送信したものとバス状態の違いにより、通信調停に負けたことを検出し送信を停止します。

このことより、複数ノードから同時送信が起こってしまった場合、他ノードがリセッシブ送信を行っているときにドミナントを送信したノードが通信調停に勝つので、優先順位が高い **ID** は **0x0** であり、**ID** の値が小さいものほど優先順位が高くなります。

ID の割り当てについてはデータフレームのデータフィールドの割り当てと同様に、設計者が自由に割り当てできますが、通信調停時の優先順位の関係で重要度の高いものは **ID** の値を小さくするなど、ネットワーク全体を考慮する必要があります。

基本的には **ID** のみで通信調停が行えますが、なぜ **RTR** も通信調停に使用するのでしょうか？ それは、同じ **ID** のデータフレームとリモートフレームが同時に送信された場合、**ID** のみでは調停は行えないからです。このようなことが起きた場合、**ID** だけではなく **RTR** も使用して通信調停を行います。データフレームでは **RTR** はドミナント、リモートフレームでは **RTR** はリセッシブであるため、データフレームが優先されることとなります。

6章2節 LIN (Local Interconnect Network)

6. 2. 1. LIN 策定の背景

近年、自動車に対する安全性や利便性の向上と環境規制への対応に伴い、電子制御部品が増加しています。この自動車の「電子制御化」は、エンジンなどを制御するパワートレイン制御やステアリングなどを制御するシャシー制御だけではなく、パワーウィンドウやミラー調整、電動シート、ドアロックなどのボディー制御に使われる「サブネットワーク」にも広がっています。

また一方で、こうしたサブネットワークの電子制御化の流れは、センサー、アクチュエータ、それらを制御する ECU (Electronic Control Unit) などの部品、そして、配線 (ハーネス) の増加をもたらしています。(下図) 当然、電子制御化に伴う配線数の増加は、材料費や開発費、組み立て工数などのコストにはね返りますし、重量や配線スペース、接触不良などの電氣的トラブルの増加など、自動車の品質と信頼性に大きな影響を及ぼすこともあり得ます。

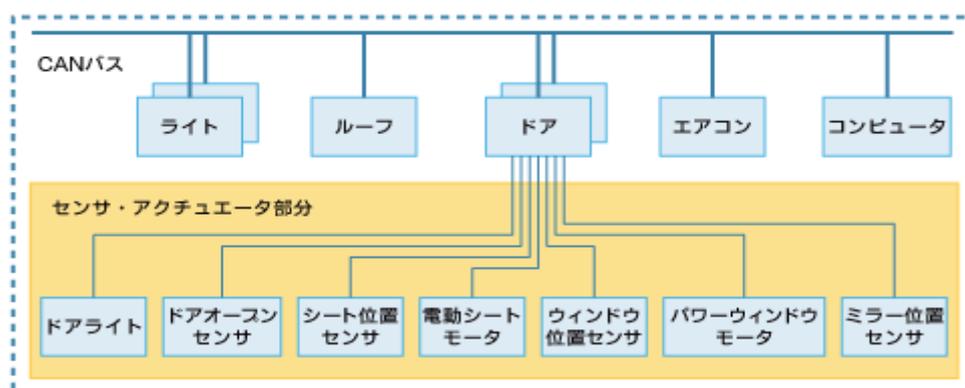


図 88 配線数の増加 (出典：ベクター)

こうした課題の解決策の 1 つに「多重通信プロトコル」の導入があります。多重通信プロトコルを導入することで、配線数の増加を抑えながらも、センサーとアクチュエータの増加に対応することが可能になります。

自動車における多重通信プロトコルといえば、CAN (Controller Area Network) をイメージするでしょう。CAN は、すでに多くの自動車に搭載されている実績があり、採用しやすい手段です。

しかし、センサーやアクチュエータなどのサブネットワーク通信に、パワートレイン制御やシャシー制御に求められる通信速度や信頼性は必要ではなく、CAN を採用することはコスト面から見ても必ずしも最適な設計とはいえません。

そこで策定された通信プロトコルが LIN (Local Interconnect Network) です。パワートレイン制御やシャシー制御ほど通信速度、信頼性を必要としないセンサーやアクチュエータなどの制御、すなわちボディー制御に採用され、シンプルかつ安価な車載向けサブネットワークシステムを構築できます。

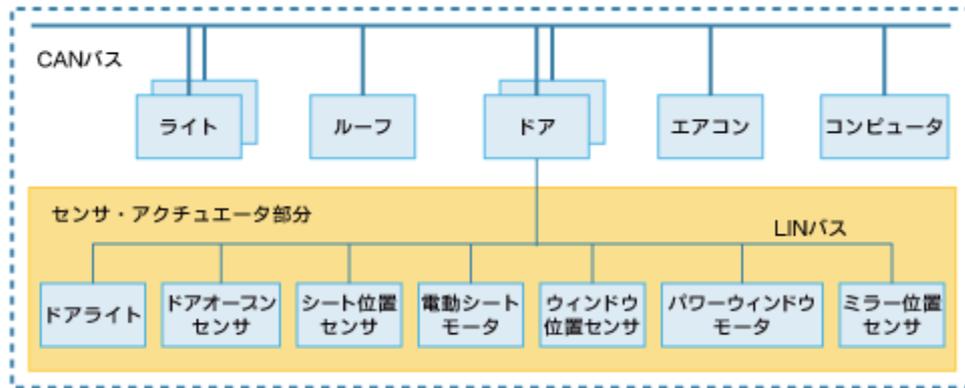


図 89 LIN による配線の減少効果 (出典：ベクター)

LIN の適用分野は、ドアミラー、パワーシート、サンルーフ、ドアロック、エアコン、照明など、特に快適性の機能分野で使用されています。

6. 2. 2. LIN の特徴と適応分野

LIN は、ドアミラー、パワーシート、サンルーフ、ドアロック、エアコン、照明など、特に快適性の機能分野で使用されています。

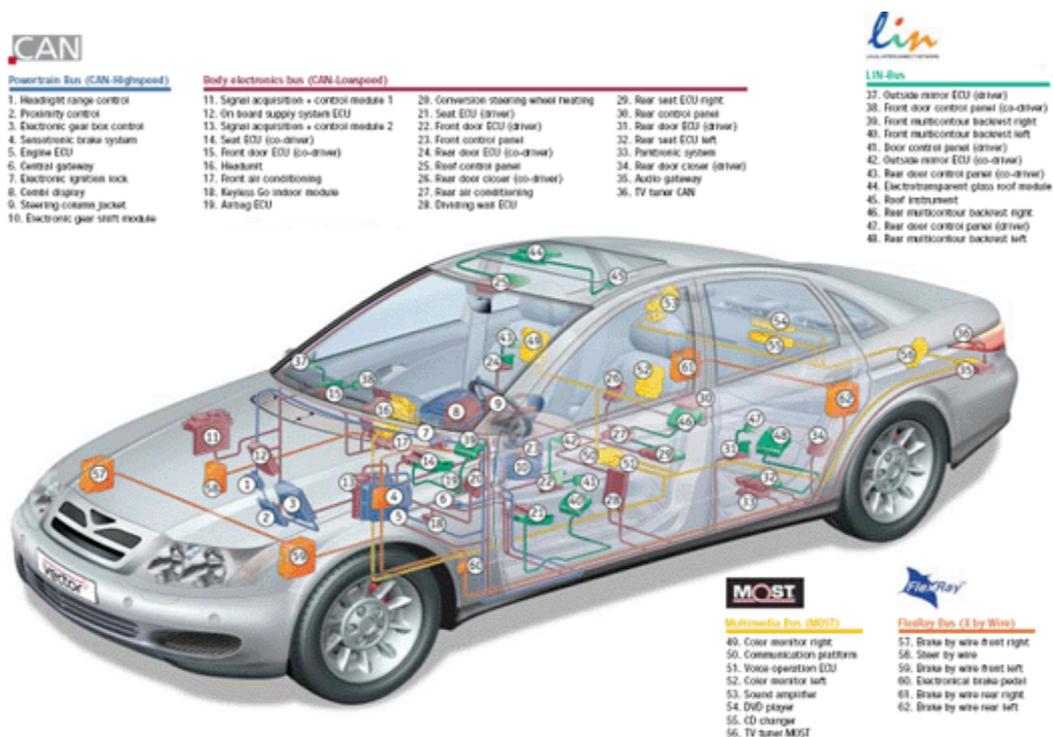


図 90 車載ネットワークの適用分野 (出典：ベクター・ジャパン)

LIN プロトコルの主な特徴は、以下のとおりです。

- ◇ライン型バス構造
- ◇マスター・スレーブ方式 (1つのマスターと複数のスレーブ)
- ◇トークン方式
- ◇メッセージ・アドレッシング方式
(すべてのノードが LIN メッセージを受信可能)
- ◇スケジュールに基づく通信 (タイムトリガー方式)
- ◇データ転送速度 最大 20kbit/sec
- ◇UART インターフェース (多くのマイコンに実装されているシリアル通信装置)
- ◇水晶発振子、セラミックレゾネータを必要としない自己同期 (スレーブノード)
- ◇短いメッセージ (最大 8 バイト)
- ◇簡単な送信データ保護 (パリティ、チェックサム)

今後も、自動車は快適装備品の増加や品質向上、コスト削減といった要求に対応していく必要があります。また、「HV (Hybrid Vehicle)」「EV (Electric Vehicle)」などの開発では、従来とは異なる制御、通信も増加していくことが考えられます。これらの要求に対応するために、LIN が使われるケースはさらに増えると推測されます。

6. 2. 3. LIN ハードウェア

LIN ノードは、マイクロコントローラ (マイコン) と LIN トランシーバによって構成されています。LIN は、簡単・安価にセンサーとアクチュエータとを接続するために、多くのマイコンを搭載しているシリアル通信装置 UART (Universal Asynchronous Receiver/Transmitter) を使用して送受信を行います。

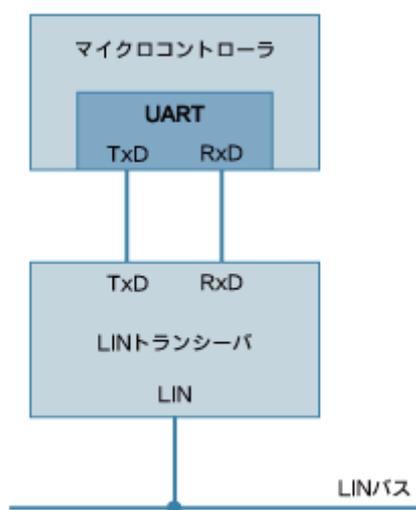


図 91 LIN ノードの構成 (出典 : monoist)

UARTの通信方式をLINで使用する場合、8ビットのデータの前後に「スタートビット」と「ストップビット」を付与した10ビット単位で送信します。また、8ビットのデータは、最下位ビット（LSB）から送信します。

スタートビットとストップビットは、データの開始と終了を判断するために使われます。スタートビットは論理値「0」となり、電圧レベルはグラウンドです。ストップビットは論理値「1」となり、電圧レベルはバッテリーです。論理値「1」を「リセッシブ」、論理値「0」を「ドミナント」と呼びます（下図）。

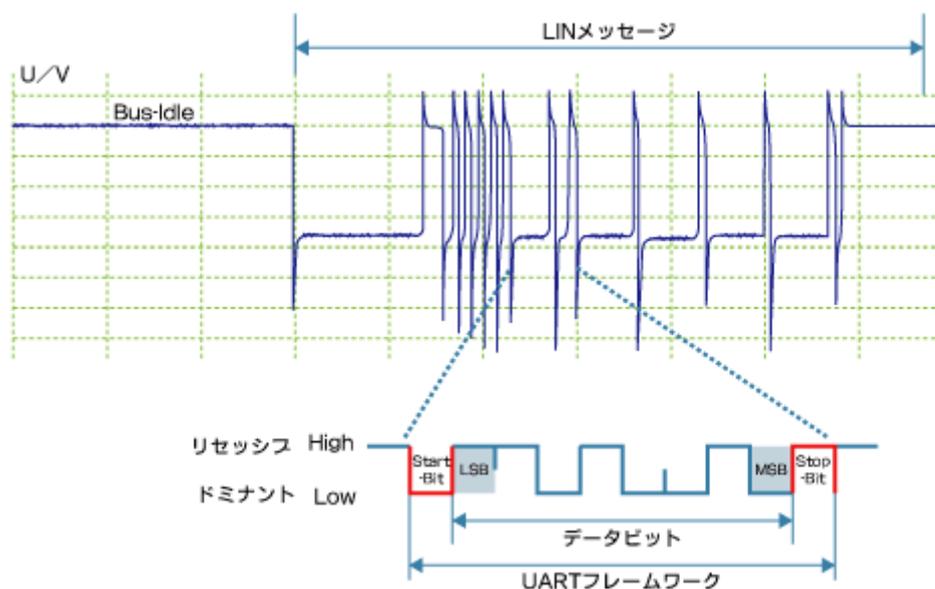


図 92 LIN のデータ構造（出典：monoist）

LIN トランシーバは、基本的に「ISO 9141」に準拠したシングルワイヤを使用します。LIN トランシーバでは入力電圧、信号の振幅、LIN ノードの省電力のためのスリープ・ウェイクアップ機能を制御します。最近では、LIN プロトコル用にカスタマイズされた「LIN UART」なども増えています。

通信速度は EMC 対策やクロック同期のために、最大 20kbit/sec と規定されています。一般的に 2.4kbit/sec、9.6kbit/sec、19.2kbit/sec がよく使われています。

このように、CAN と比較すると通信コントローラを必要とせず、マイコンに搭載されている UART で通信できるので、ハードウェアのコストを抑えることができます。また、LIN バスはシングルワイヤを使用するため、配線数を抑え、コストを削減することができます。

6. 2. 4. 通信方式

LIN の通信は、

◇マスター・スレーブ方式

◇スケジュールに基づく通信（タイムトリガー方式）

で行われます。これは CAN とは異なる通信方式で、LIN の大きな特徴といえます。

「電車」を例にして、LIN の通信方式の特徴を説明します。

電車は、始発駅から時刻表どおりに出発します。この時刻表には、電車が衝突しないように出発時刻が決められています。また、駅では乗客が電車を待ち、乗車対象の電車が到着すると乗車し、目的の駅まで移動します。

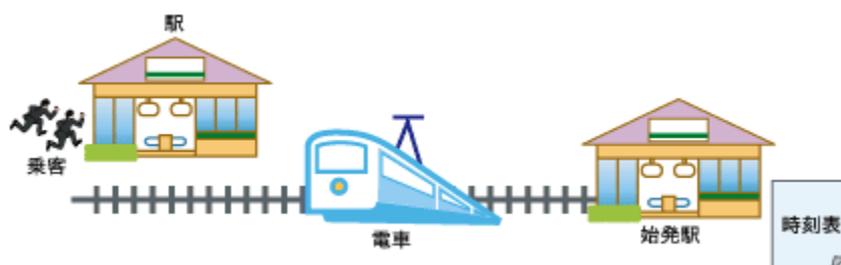


図 93 電車を例に LIN の通信方式を説明（出典：monoist）

LIN の通信も、この例と同様の動作をします。LIN では、事前に定義された「送信タイミング」に従って送信を行います。そのため、メッセージの衝突は発生せず、各ノードは一定の間隔で確実にメッセージの送信および受信ができます。また、バスも過負荷にならないので安定した通信ができます。

しかし、送信するタイミングが決められているということは、各ノードは任意のタイミングでメッセージを送信できません。送信するタイミングが来るまで待つ必要があります。

また、ネットワーク内に「送信するタイミングを制御する」という特別な役割を実行するノードが必要となるため、LIN では「マスターノード」と「スレーブノード」といった 2 種類のノードを使用して通信を行います。

CAN では、各ノードは任意のタイミングで送信できますが、メッセージの衝突の調停によっては必ずしも周期性が保証できないなど、あらかじめ期待されたタイミングで確実にメッセージのやりとり（通信）ができないことがあります。しかし、LIN では送信するタイミングをあらかじめ定義することにより、期待されたタイミングで確実にメッセージの通信が行えます。

6. 2. 5. ライン型バス構造

LIN のネットワーク構造（トポロジー）は、「ライン型バス構造」です。1 つの LIN バスラインに 1 つのマスターノードと複数のスレーブノードが接続できます。また、LIN ネットワークの推奨最大ノード数は 16 ノード、最大配線長は 40 メートルです。

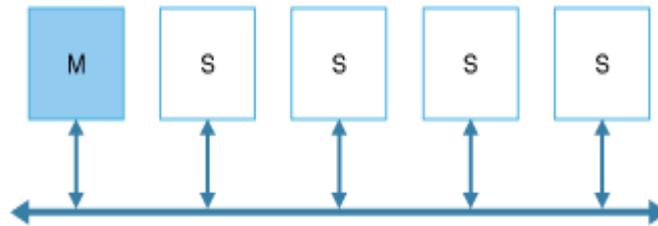


図 94 LIN のネットワーク構造（出典：monoist）

6. 2. 6. マスター・スレーブ方式

LIN ネットワークは、マスターノードが全体の通信を制御するマスター・スレーブ方式を採用しています。スレーブノードは、マスターノードに従って通信を行います。

LIN ノードの役割には、「マスタータスク」と「スレーブタスク」の 2 種類があります。

〔1〕 マスタータスク

マスターノードのみが持っている役割で、トークンの送信とスケジュールの管理を行います。これは、決められたタイミングで LIN バスに送信要求「トークン」を送信します。

〔2〕 スレーブタスク

マスターノードとスレーブノードの両方が持っている役割で「データ」を送信します。データの送信は、マスタータスクから送信されるトークンをスレーブタスクが監視し、データを送信するノードがトークンの後にデータを送信します。つまり、マスターノードからトークンが送信されない限り、各ノードはデータを送信できません。

LIN では、このトークンとデータで 1 つの「メッセージ」を構成しています。

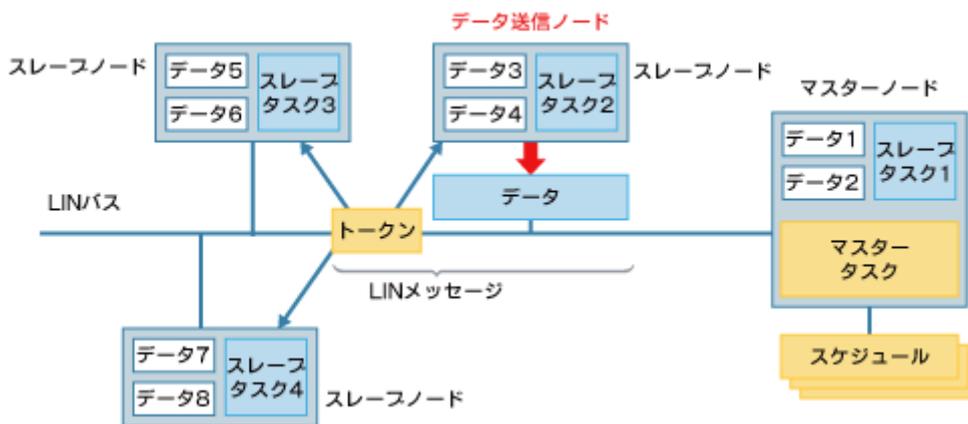


図 95 マスター・スレーブ方式 (出典 : monoist)

※各ノードは、マスターノードから送信されるトークンを監視し、データ送信ノードがトークンの後にデータを送信。また、マスターノードはスレーブタスクを持っているため、データの送信もできます。

LIN ではトークンを「ヘッダー」、データを「レスポンス」、メッセージを「フレーム」と呼んでいます。ヘッダーには、フレームの意味を表す「ID」と呼ばれる情報を持っており、各ノードはこの ID を監視することでレスポンスを送信するかどうかを識別します。また、LIN フレームは「メッセージ・アドレッシング方式」で送信されるため、1つ、複数、すべての LIN ノードが LIN フレームを受信できます。

6. 2. 7. スケジューリング

マスタータスクは、ヘッダーを送信するタイミングを「LIN スケジュール」で定義します。LIN スケジュールには送信する ID や送信の順番、送信する時間間隔が定義されています。

つまり、マスターノードはこの LIN スケジュールを使用して、LIN ネットワーク全体の通信の制御を行います。そのため、LIN スケジュールは LIN バス上で通信の衝突が発生しないように送信タイミングを定義する必要があります。

また、LIN スケジュールは複数のスケジュールテーブルの定義ができます。つまり、マスターノードは「起動時の初期化モード」「通常モード」「診断モード」など、車両状態の変化に応じてスケジュールの変更を行い、送信 ID や送信周期の変更ができます。

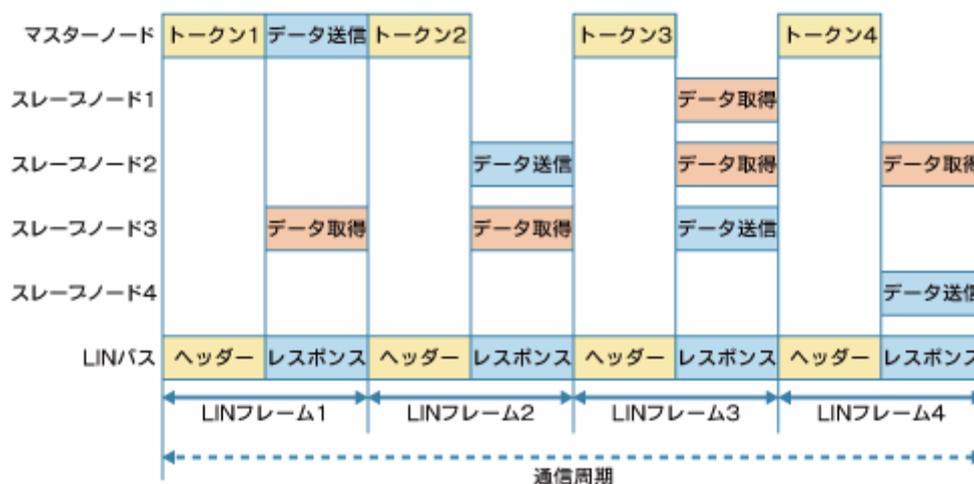


図 96 LIN スケジュール (出典 : monoist)

※LIN フレーム 1 では、マスターノードがデータを送信し、スレーブノード 3 がデータを取得。LIN フレーム 3 では、スレーブノード 3 がデータを送信し、スレーブノード 1、2 がデータを受信。LIN スケジュールの最後のフレーム (LIN フレーム 4) を送信すると、最初 (LIN フレーム 1) の送信に戻ります。

6. 2. 8. 同期方法

LIN ノードはコスト削減のため、各信号の通信速度を調整する専用の配線などを使用していません。また、LIN スレーブノードのクロック回路もコストを抑えるために、誤差が大きい CR 発振やリングオシレーター発振などの使用が許されています。つまり、そのままでは、LIN ノード間の内部クロックに誤差が生じる可能性があります。

そこで、LIN ではこの“クロックの誤差”を補正するために、マスターノードから送信されるヘッダーを使用します。

LIN では、マスターノードは水晶発振子などの高精度な受動素子の使用が規定されており、その許容誤差は±0.5%です。これにより、マスターノードは正確なクロックの作成ができます。この高性能な内部クロックを持つマスターノードから送信されるヘッダーに、クロック誤差を補正するための「同期信号」を入れることにより、各ノード間のクロック誤差を補正します。

※LIN ではスレーブノードのクロックの許容誤差は±14% (LIN1.3 では、±15%) と規定されています。ただし、高精度のクロック回路を使用した場合は、許容誤差±1.5%と規定されています。

6. 2. 9. フレーム構造

1 つのフレームは、ヘッダーとレスポンスで構成されていて、ヘッダーはマスタータスク（マスターノード）から、レスポンスはスレーブタスク（マスターノードまたはスレーブノード）からそれぞれ送信されます。

〔1〕ヘッダー

ヘッダーは「Break」「Synch」「Protected ID (PID)」の 3 つのフィールドで構成されています。

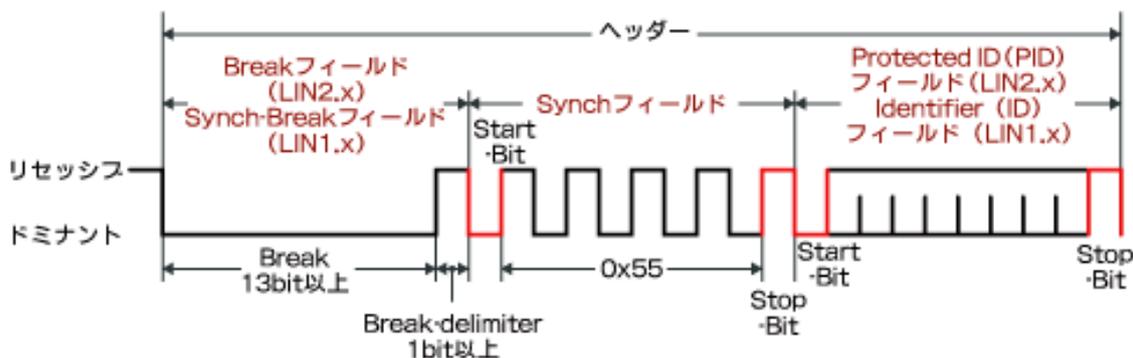


図 97 ヘッダーの構成（出典：monoist）

※LIN1.x は、プロトコルバージョン LIN1.0～1.3、LIN2.x は LIN2.0～2.1 を表す。Break フィールドと PID フィールドは、LIN プロトコルバージョンによって名称が異なります

①Break フィールド

Break フィールドは、ほかのフィールドと異なり、意図的にフレーミングエラー（スタートビットから数えて 10 ビット目にストップビットが検出されない場合に発生）を起こすことにより、すべてのスレーブノードに LIN フレームの開始を通知します。

Break フィールドは Break と Break-delimiter で構成され、Break は 13 ビット以上のドミナント、Break-delimiter は 1 ビット以上のリセッシブとなります。Break-delimiter は、Break の終わりを表します。

②Synch フィールド

同期方法で説明した「同期信号」が Synch フィールドです。Synch フィールドは、各ノード間のクロック誤差の補正に使用し、0x55 UART フレーム（スタートビット／ストップビットあり）を送信します。スレーブノードは、立ち下がりエッジの最初と最後の時間差を 8 で割ることによって算出された 1 ビットの時間間隔 (TBit) を基に、各ノードのクロック誤差を調整します（下図）。ただし、高精度のクロック回路（許

容誤差±1.5%) を使用した場合は、クロック誤差の調整は必要ありません。

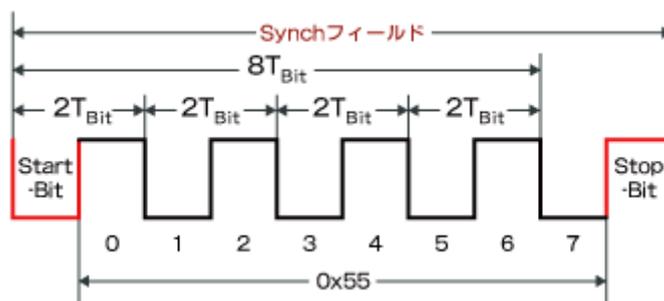


図 98 Synch フィールド (出典 : monoist)

Synch フィールドにより、各ノードの通信速度をマスターノードの通信速度に合わせるにより各ノードのクロック誤差が調整されます。

③Protected ID (PID) フィールド

PID フィールドは、LIN フレームの識別情報を表す 6 ビット (0~5 ビット) のフレーム ID と、2 ビット (6、7 ビット) のパリティの合計 8 ビットで構成されています (下図)。

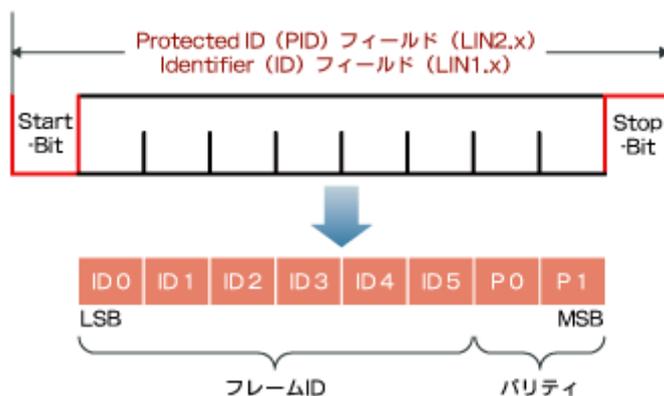


図 99 Protected ID フィールド (出典 : monoist)

※1 フレーム ID の範囲は、6 ビットで表現できる 0~63 (0x3F) です。ただし、フレーム ID 60 (0x3C)、61 (0x3D) は診断フレームとして使用するなど、LIN 仕様上、すでに役割が定義されているフレーム ID もあります。

※2 パリティには、フレーム ID から演算式によって求められた値が格納されます。ヘッダーを受信したノードは、ヘッダー内のパリティビットの値とフレーム ID から演算した値を比較し、正しいフレーム ID を受信したかどうかを確認します。

<p>※パリティビット(P0、P1)の演算式</p> <p>$P0 = ID0 * ID1 * ID2 * ID4$</p> <p>$P1 = \neg (ID1 * ID3 * ID4 * ID5)$</p> <p>*: XOR ¬: INV(反転)</p>	<p>例</p> <p>ID = 0x22 = 10 0010b</p> <p>$P0 = 0 \oplus 1 \oplus 0 \oplus 0 = 1$</p> <p>$P1 = \neg(1 \oplus 0 \oplus 0 \oplus 1) = 1$</p> <p>PID = 1110 0010b = 0xE2</p>
---	---

図 100 パリティビット演算式 (出典 : monoist)

LIN1.x では、PID フィールドを「Identifier (ID)」フィールドと呼び、5、6 ビット目 (上の図の ID4、ID5) にレスポンスのデータ長 (DLC : Data Length Code) の情報を格納する場合があります。

〔2〕レスポンス

レスポンスは「データ」「チェックサム」の 2 つのフィールドで構成されています。データ、チェックサムともに、UART フレーム (スタートビット/ストップビットあり) で送信します。

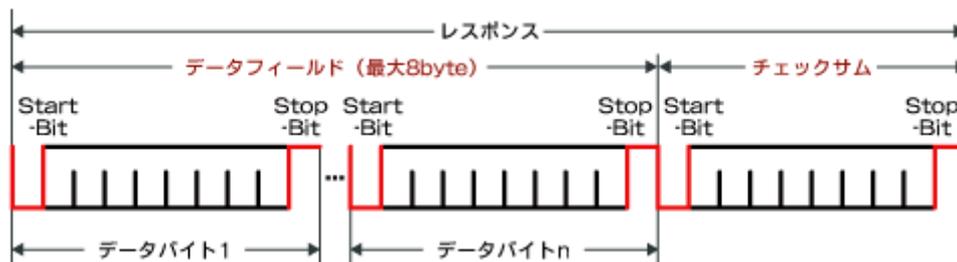


図 101 レスポンス (出典 : monoist)

①データフィールド

データフィールドには、最大 8 バイトのデータが格納されます。

②チェックサム

チェックサムは、データを正確に受信できたかどうかを確認するために使用します。チェックサムには、各データ値の総和を反転した値が格納されます。ただし、総和の結果がけたあふれとなった場合、けた上がり値を演算結果に加算する必要があります (モジュロ 256 方式)。チェックサムには、「標準チェックサム (Classic Checksum)」 「拡張チェックサム (Enhanced Checksum)」の 2 種類があります。

A. 標準チェックサム

- 演算対象は、すべてのデータバイト
- LIN1.x のすべてのフレーム ID に使用
- LIN2.x では、診断フレーム（フレーム ID 60～61）のみ使用

Data0	A0	8ビット値	58	標準
Data1	+ B8	桁上がり値を加算	+ 01	チェックサム = A6
Sum1	158	Sum2	59	
		INV	A6	

図 102 標準チェックサムの例（出典：monoist）

B. 拡張チェックサム

- 演算対象は、PID およびすべてのデータバイト
- LIN2.x のフレーム ID 0～59 に使用

PID	E2	8ビット値	82	8ビット値	3B	拡張
Date0	+ A0	桁上がり値を加算	+ 01	桁上がり値を加算	+ 01	チェックサム = C3
Sum1	182	Result1	83	Result2	3C	
		Data1	+ B8	INV	C3	
		Sum2	13B			

図 103 拡張チェックサムの例（出典：monoist）

このように、LIN フレームの先頭部分（Synch フィールド）でクロック誤差の調整をすることで、各スレーブノードはマスターノードからフレーム ID を受信し、データの送受信を行うことができます。また、フレーム ID やデータが正しく送受信できたかどうかを各ノードが確認するために、LIN フレームにはパリティ、チェックサムが含まれています。

6. 2. 10. 時間規定

LIN フレームは、LIN スケジュールによって各フレームの送信タイミングが重複しないように定義されます。その簡易的なハードウェア構成により、LIN フレームの送信時間に「レスポンススペース」や「インターバイトスペース」と呼ばれる時間が含まれることを許容しており、これらの時間誤差を考慮してスケジュールを設計する必要があります。

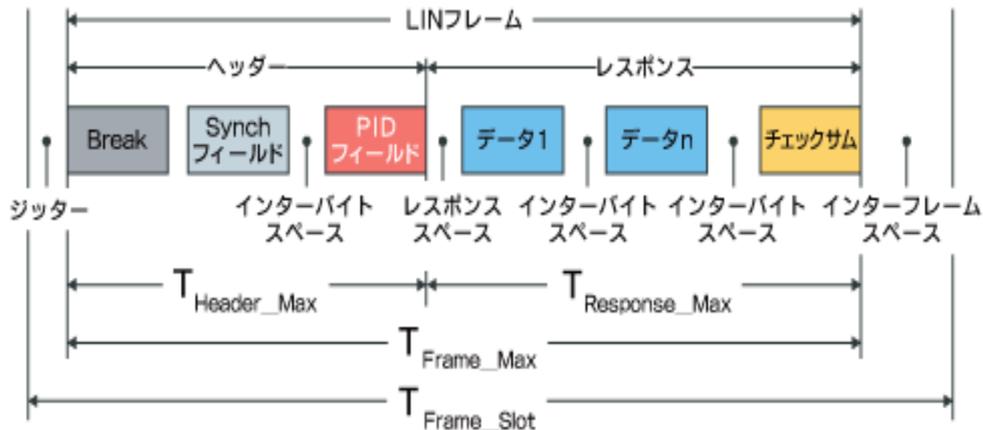


図 104 時間規定 (出典 : monoist)

LIN フレームの送信に必要な時間は、以下の式で求めることができます。「公称タイミング」とはレスポンススペース、インターバイトスペースを含めず、Break を 13 ビット (最小値)、Break-delimiter を 1 ビット (最小値) として求めたビット時間です。

<公称タイミング>

- $T_{Header_Nominal} = 34T_{Bit}$
Break(13TBit) + Break-delimiter(1TBit) + Synch(10TBit) + PID(10TBit)
- $T_{Response_Nominal} = (N_{Data} + 1) \times 10T_{Bit}$
Data(10TBit) × データバイト数(N_{Data}) + チェックサム(10TBit)
- $T_{Frame_Nominal} = T_{Header_Nominal} + T_{Response_Nominal}$

レスポンススペース、インターバイトスペースなどの許容時間は、LIN フレームごとに最大 40%です。そのため、LIN フレームの最大時間は、以下の式で求めることができます。

<最大タイミング>

- $T_{Header_Max} = 1.4 \times T_{Header_Nominal}$
- $T_{Response_Max} = 1.4 \times T_{Response_Nominal}$
- $T_{Frame_Max} = T_{Header_Max} + T_{Response_Max}$

LIN スケジュールの時間単位は、「タイムベース (Time_Base)」と呼ばれ、LIN スケジュールを処理するための最小時間単位となります。例えば、タイムベースが 5ms となっている場合、LIN フレームの送信間隔は 5ms 単位で設定ができます。一般的にタイムベースは、5ms または 10ms がよく使われています。

また、一つの LIN フレームに必要なタイムベースの合計を「フレームスロット (Frame-Slot)」と呼びます。

6. 2. 11. フレームタイプ

LIN のフレームタイプは、「アンコンディショナル (Unconditional)」「イベントトリガー (Event Triggered)」「スποラディック (Sporadic)」「診断 (Diagnostic)」の 4 種類となります。

〔1〕 アンコンディショナルフレーム (フレーム ID 0~59)

通常使われる LIN のフレームタイプです。LIN ノード間のデータ通信に使われます。



図 105 アンコンディショナルフレーム (出典 : monoist)

〔2〕 イベントトリガーフレーム (フレーム ID 0~59) ※LIN2.0 で追加

発生頻度の少ないイベントに使用するスレーブノード用のフレームで、1 つのヘッダーに対して複数のレスポンスを定義できます。ヘッダーは LIN スケジュールのとおり送信されますが、レスポンスは必要なときにだけ送信されます。送信したノードが判断できるように、データバイトの 1 バイト目は PID が格納されます。



図 106 イベントトリガーフレーム (出典 : monoist)

イベントトリガーフレームのフレーム ID は、関連するスレーブノードで共有しているため、同時に複数のスレーブノードがレスポンスを送信し、衝突する可能性があります。衝突が発生した場合、衝突によって失われたレスポンスデータを再送信させるために、マスターノードはアンコンディショナルフレームを使用し、スレーブノードごとに再送信要求を行います。LIN2.1 では、イベントトリガーフレームの衝突発生時に使用する「コリジョンリゾルビング (Collision Resolving) スケジュールテーブル」が追加されています。

イベントトリガーフレームの使用例として、4 ドア・セントラル・ロックング・システムでのドアロックの監視があります。イベントトリガーフレームを使用した場合、1 つのフレームスロットで 4 つのドアロックを監視できます。フレームスロットの減少は帯域の有効活用につながり、より効率的な通信を設定できます。しかし、同時に複数のドアロックが操作された場合は衝突が発生するので、再送信が必要です。もし、ドアロックの監視にアンコンディショナルフレームを使用した場合、ほとんど変化しないドアロック信号に 4 つのフレームスロットを使用するので、帯域の無駄となります。このように、発生頻度の少ないイベントに対し、1 つのフレームスロットで通信することで、帯域の有効活用ができます。

〔3〕 スポラディックフレーム (ID 0~59) ※LIN2.0 で追加

特定のシグナルが更新された場合などに使用されるマスターノード用のフレームです。1 つのフレームスロットに複数のスポラディックフレームを定義できます。ただし、同時に複数のシグナルが更新された場合は、優先順位の高いフレームを最初に送信します。シグナル更新がない場合、マスターノードはスポラディックフレームのレスポンスだけでなくヘッダーも送信しません。

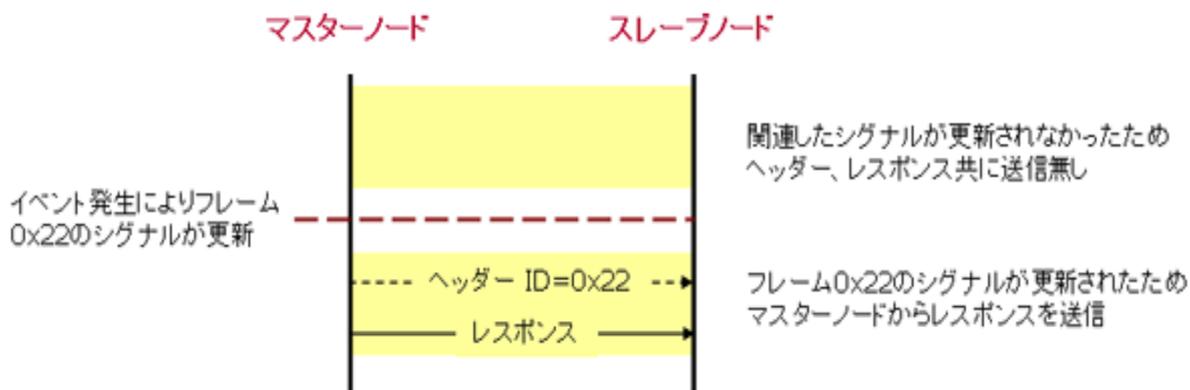


図 107 スポラディックフレーム (出典 : monoist)

6章3節 FlexRay

近年、自動車には多くの「ECU（Electronic Control Unit：電子制御装置）」が搭載されるようになり、それらがネットワークに接続されて互いに制御情報を通信することで、より高度な機能を実現しています。

この車載ネットワークではこれまで「CAN（Controller Area Network）」が標準的な通信プロトコル（規格）として採用されてきましたが、現在、より優れた高速性と信頼性を実現する技術として注目されているのが、次世代車載ネットワーク「FlexRay」です。

FlexRay は、自動車内のエレクトロニクス化が進むに従って表面化してきたさまざまな問題点（例えば通信量の増加など）を解決する技術として、またいわゆる“X-by-Wire”を実現するための基幹技術として策定された通信プロトコルであり、高い柔軟性と信頼性、高速性を特徴としています。

6. 3. 1. なぜ新しい通信プロトコルが求められているのか？

自動車の電子制御化が進むに従って、さまざまなニーズが生まれています。例えば次のような事柄です。

- ECU 数の増加に伴う、車載ネットワークの通信量の増加、複雑化への対応
- 商品性の向上、システムの最適化、コスト削減のための統合的な協調制御
- 機能や性能の向上のための X-by-Wire システムへの移行

〔1〕 車載ネットワークの通信量の増加、複雑化への対応

1997 年以前、パワートレイン（駆動系）向け車載ネットワークに接続されている ECU（通信ノード）の数は 6～8 程度でした。ところが、最近の上級車に搭載されている ECU の数は 100 近くにもなり、通信量が膨大になっています。また、ネットワークが複雑化した結果、設計の難易度が増し、品質や機能の低減も懸念されます。これら問題は従来の CAN による通信量／通信速度では解決が難しく、より多くのデータを通信できるネットワークプロトコルが必要となります。

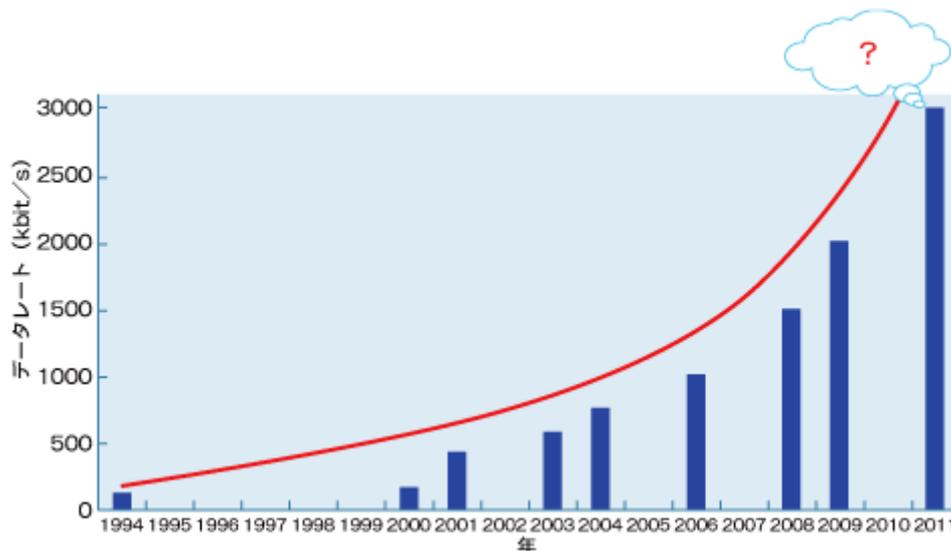


図 109 (例) 車載ネットワークにおける通信データ量の増加 (出典: monoist)

〔2〕 商品性の向上、システムの最適化、コスト削減のための統合的な協調制御

これまで、各 ECU は個別に機能分担されて制御されてきました。しかし、近年はさらなる商品性の向上、システムの最適化、コスト削減のため、分散配置された ECU 同士がネットワークを介することで車両を統合的に協調制御するシステムが求められています。

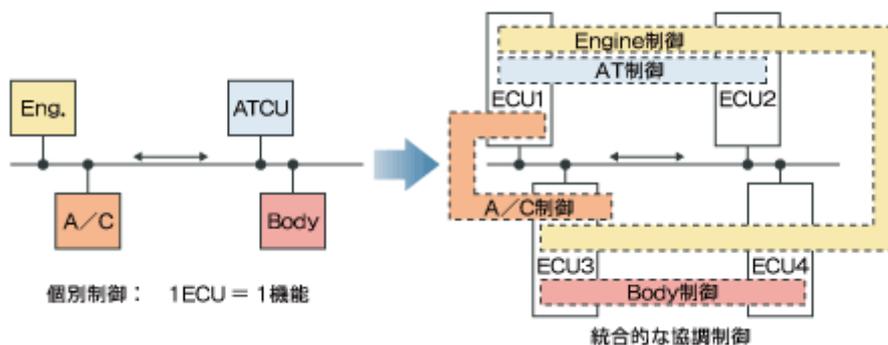


図 110 個別制御から協調制御へ (出典: monoist)

例えば、上級車に採用されている「プリクラッシュ・セーフティ・システム」と呼ばれるシステムがあります。これは、レーダーやカメラの情報を基に、先行車や障害物、歩行者などとの衝突を予測し、ドライバーへの警告、衝突の回避、または衝突が避けられないと判断したときはその被害の軽減を図るシステムです。ここではレーダーやカメラなどのセンサー、ブレーキシステム、メータパネル表示システムなど複数の分散したシステムが協調動作することで1つの機能を実現しています。

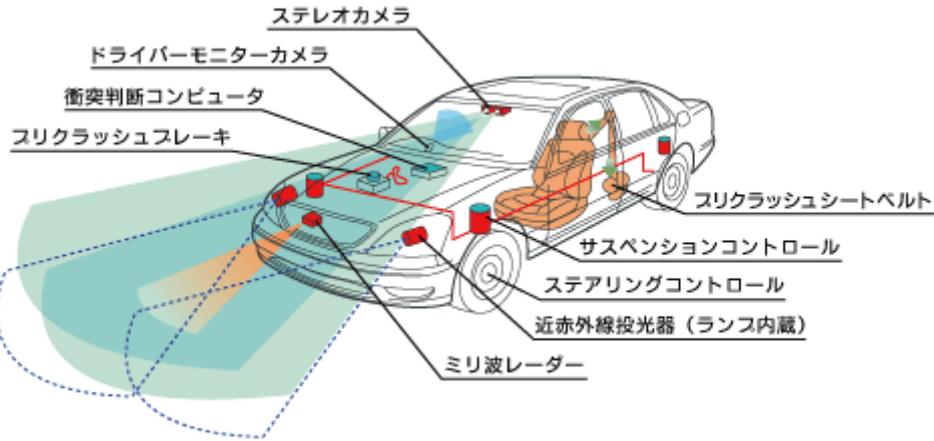


図 111 プリクラッシュ・セーフティ・システムのイメージ (出典：monoist)

このようなシステムでは、各 ECU ができるだけ同期して通信することが求められます。一方、CAN での通信フレームは送信要求が発生したときに送信され、その送信時間が必ずしも決まっていません。フレームが同時に送信され、フレーム同士の衝突が発生する場合もあり、これは送信遅延につながります。つまり、その送信遅延は通信の負荷状況（ネットワークの混雑状況など）に依存するため、協調制御では課題があります。一部必要なデータが「ネットワークが混んでいたため遅れました……」となっては都合が悪いのです。

〔3〕機能や性能の向上のための X-by-Wire システムへの移行

X-by-Wire は、機械的な伝達機構で動作している自動車の操作部分をエレクトロニクス部品に置き換える、自動車の構造そのものを根本的に変える技術です。操作部分とアクチュエータを電氣的に接続すれば済むため、機械的な部品が大幅に削減され、自動車の設計やレイアウトの自由度は大幅に高まります。例えば、X-by-Wire がステアリングやブレーキに導入された場合、コラムシャフト、油圧機構、パーキング用のケーブルなどの伝達機構が必要なくなります。また、これらを電子制御化することにより、他機構との統合制御に組み込むことがより容易になり、安全性や快適性の大幅な向上が期待できます。

この X-by-Wire を実現する際、高い信頼性／通信速度を持つ通信仕様が求められます。例えば、通信不良のためにステアリングやブレーキが正常に動作しなければ重大な事故につながってしまいます。CAN では信頼性、通信速度、ともに十分とはいえません。

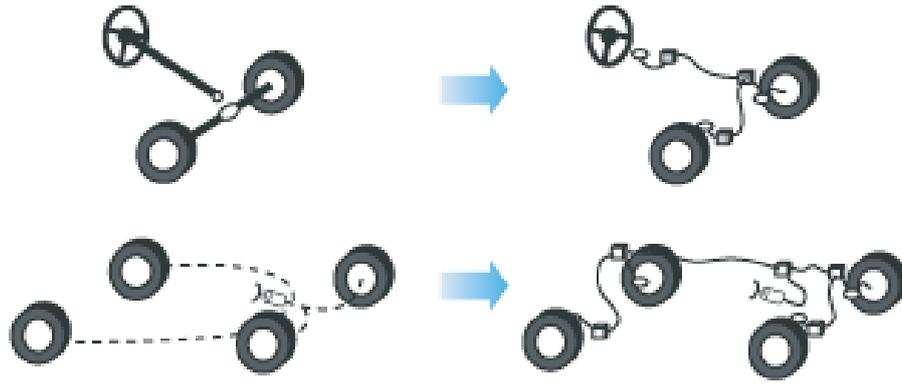


図 112 ステアリング by Wire (上) とブレーキ by Wire (下) (出典 : monoist)

これらニーズを整理すると、次世代車載通信プロトコルに求められることは次のようになります。

- ① 速い通信速度
- ② 送信遅延時間が保証されていること (決定論的な送信)
- ③ 同期通信
- ④ 高い信頼性
- ⑤ 障害耐性
- ⑥ 高い柔軟性

FlexRay は、これらの要件を満たすことができるため、X-by-Wire に対する有力な候補の 1 つとなっています。

6. 3. 2. これまでの経緯

欧州における高信頼性、高速性を備えた通信プロトコルの研究開発は 1980 年代より行われており、一時期、TTP/C (Time Triggered Protocol for SAE Class. C Applications) という通信プロトコルが EU プロジェクトにも取り上げられるほど有望視されていました。しかし、この TTP/C では信頼性を重視するあまり、コスト面や柔軟性を気にする自動車メーカーの意見を十分に反映されず、プロジェクトメンバーであった Daimler 社や BMW 社が離脱。2000 年、標準化団体 FlexRay コンソーシアムを設立し、BMW 社独自の通信プロトコル Byteflight (バイトフライト) を参考にして FlexRay 仕様の策定を進めました。

その後、次期車載通信ネットワークの標準の座をめぐり FlexRay と TTP/C は競合関係にありましたが、2005 年、Volkswagen 社など TTP/C 陣営の主要メンバーが FlexRay コンソーシアムに加入したことで、FlexRay に一本化されました。

そして2006年、BMW社がSUV車「X5モデル」にて世界で初めてFlexRayを採用しました。以来、BMW社「7シリーズ」「5シリーズ」モデル、Audi社「A8」モデルなど、徐々に採用車種が広がり、現在に至っています。

6. 3. 3. FlexRay コンソーシアム

通信プロトコルを1企業で業界標準化することは難しいため、前述のとおり、複数の企業が標準化団体（FlexRay コンソーシアム）を2000年に結成し、通信プロトコルの策定を進めました。このFlexRay コンソーシアムには計7社のコアメンバー（BMW社、Bosch社、Daimler社、Freescale Semiconductor社、General Motors社、NXP Semiconductors社、Volkswagen社）を中心に大手自動車メーカー、半導体メーカー、ツールメーカーなど計100社以上が加盟しました。

コンソーシアム活動は、2009年末に終了しています。標準化団体活動の成功例として評価されており、「AUTOSAR」などほかの標準化団体のベンチマークともなっています。

なお、2014年11月時点でのAUTOSARによるSpecification of FlexRay Interfaceのバージョンは3.3.0です。

6. 3. 4. FlexRay プロトコルの特徴

まず、FlexRay プロトコルの主な特徴は以下のとおりです。

- [1] 通信方式：TDMA（Time Division Multiple Access／時分割多重アクセス）。Fixed TDMA と Flexible TDMA の2種類に対応
- [2] ネットワークトポロジー：バス型、スター型、それらの混合型など、さまざまなトポロジーに対応
- [3] フォールトトレランス（障害耐性）：ネットワークの二重化による冗長性。可能な限り通信を継続するコンセプト
- [4] ノード構造：ホスト・通信コントローラ・バスドライバ・バスガーディアンによる相互監視
- [5] 優れた通信同期機能／共通の時間軸：グローバルタイムとさまざまな補正機能
- [6] 高速通信：最大10Mbps

〔1〕 Fixed TDMA 通信／Flexible TDMA 通信

TDMA（時分割多重アクセス）は、通信時間を一定時間ごと（「タイムスロット」もしくは「スロット」と呼ばれます）に分割することで、多重化通信を実現する「タイムトリガー」通信方式です。

FlexRay ではこの TDMA 通信方式を採用し、各フレームの送信タイミングや順番は事前に定義されます。よって、フレーム同士の衝突は起こらず、また特定のフレームによってネットワークが独占されることもなくなるため、期待どおりのタイミングで通信を行い、通信負荷をある一定以内に保つことができます。

一方、CAN のように何らかの事象=イベントの発生によってフレーム送信が行われる通信方式（イベントトリガー方式：CSMA/CA）の場合、フレームの衝突や調停（衝突したフレームのどちらを先に送信させるかを調整すること）によって、常に期待したタイミングで通信を行えるとは限りません。

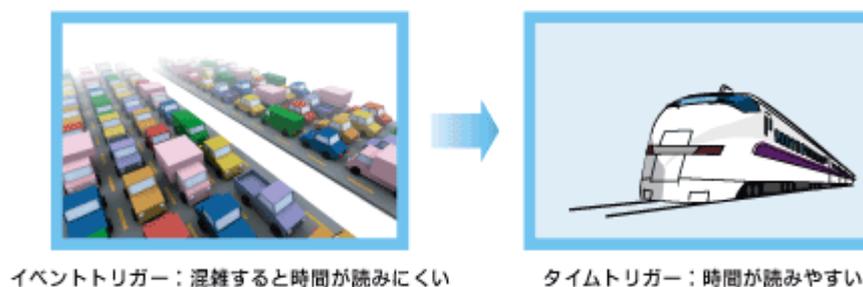


図 113 （左） イベントトリガー方式と、（右） タイムトリガー方式のイメージ
（出典：monoist）

また、FlexRay の TDMA には「Fixed TDMA」と「Flexible TDMA」の 2 種類があることが大きな特徴です。

① Fixed TDMA 通信

◇決まったフレームを決まったタイミングで必ず送信

（=最大送信遅延時間を保証）

◇送信タイミング（スロットの長さ）は、送信フレームの有無にかかわらず一定

◇安全性重視のアプリケーションに適している

② Flexible TDMA 通信

◇決まったフレームを要求に基づき任意のタイミングで送信（=高応答性）

◇送信「順番」は、事前に定義（フレームの衝突は発生しない）

◇送信タイミング（スロットの長さ）は、送信フレームの有無によって変わる

◇高い応答性が求められるアプリケーションに適している

それぞれの特徴を生かすことにより、一定の送信周期を保証する、もしくは迅速なレスポンスを実現するなど、アプリケーション志向の通信を柔軟に設計できます。(FlexRayの「flex」はFlexibility<柔軟性>から由来しているといわれています。)

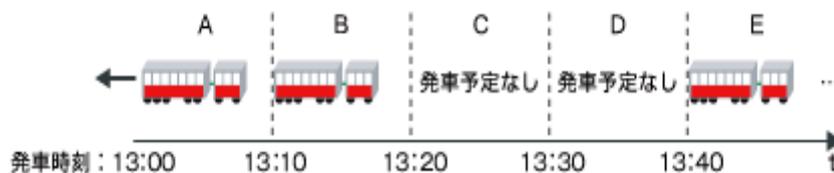


図 114 Fixed TDMA 通信の例 (出典: monoist)

- ・発車予定やその時間は事前に決まっている
- ・電車の長さや発車間隔は一定

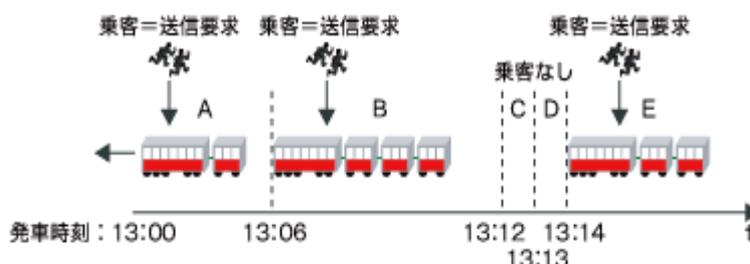


図 115 Flexible TDMA 通信の例 (出典: monoist)

- ・発車時間は乗客次第。順番のみ事前に決まっている
- ・電車の長さも一定ではない

〔2〕 ネットワークトポロジー

ネットワークトポロジー (構成) は、「バス型」から「パッシブ・スター型」「アクティブ・スター型」および、それらの「混合 (ハイブリッド) 型」まで、多様な形が可能です。これにより、自由度の高い車載ネットワーク設計ができます。

バス型は、CAN などにも採用されている一般的なトポロジーです。コスト効率が良い反面、バスの長さは主に電気信号の反射と伝播遅延により制限され、ネットワーク上のある個所で発生した障害によりネットワーク全体が影響を受ける可能性が大きい、といった懸案もあります。

スター型には“パッシブ”と“アクティブ”の2種類があります。パッシブ・スター型は、基本的にバス型と同じ考え方です。一方、アクティブ・スター型は「スター・カップラー」と呼ばれるゲートウェイのような役割を行う装置に各ノードが接続されたトポロジです。アクティブ・スター型では、バス長をより長くすることができるため、大きいネットワークを構築できます。また、ネットワーク上のある個所で障害が発生した場合のネットワーク全体に与える影響度も抑えることができるため、より安全重視のアプリケーション向きといえます。

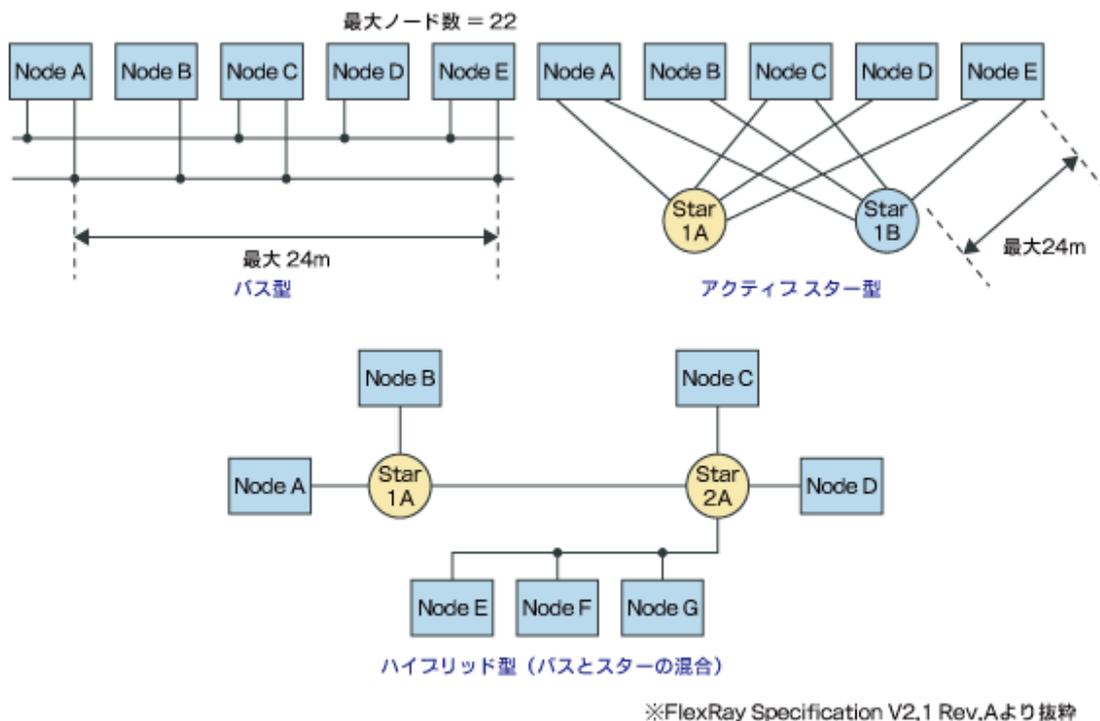


図 116 FlexRay のネットワークトポロジ (出典 : monoist)

〔3〕フォールトトレランス (障害耐性)

ネットワークの二重化 (A チャンネル/B チャンネル) に対応することで、冗長性を持たせています。つまり、仮に一方のネットワーク線に断線などの障害が発生しても、残りのネットワーク系統により正常な通信が維持されることとなります。ただし、この二重化は必須ではなく、要求される信頼性やコストに従って、ネットワークごと、ノードごとに柔軟に設計できます。

また、FlexRay では可能な限り通信を継続するコンセプトに基づいているため、エラーによって通信を中断する条件は通信プロトコルでは規定しておらず、アプリケーションで判断します。つまり、致命的なエラーを検出した場合も、通信コントローラが停止状態 (halt) には遷移しますが、プロトコルが「勝手に」通信を中止させることはしません。その決定はアプリケーションで行われるこ

とになります。より安全性、信頼性が求められるシステムに適した仕様といえます。

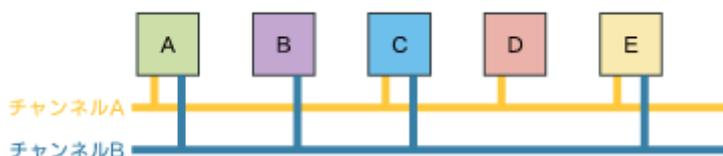


図 117 ネットワーク二重化の例（出典：monoist）

ノード A、C、E は 2 つのチャンネルに接続。ノード B はチャンネル B へのみ、ノード D はチャンネル A のみに接続しています。

〔4〕 ノード構造

FlexRay ノードの構造は、複数の「論理ブロック」と呼ばれる機能体（以下）から構成され、それぞれが多岐にわたる情報を共有し、相互監視を行っています。

①ホスト（マイクロコントローラ）：

アプリケーションの一部として、通信コントローラにアプリケーションの制御情報、設定情報、データを提供することや、バスドライバの動作モードを制御することを担当します。1 ノードに 1 つ装備されます。

②通信コントローラ（CC）：

通信プロトコルの処理を担当します。ホストへシステム状態の情報、受信したデータを提供し、バスドライバに送信データや送信許可信号を提供します。1 ノードに 1 つ装備されます。

③バスドライバ（BD）：

チャンネルへの物理的なアクセスを担当します。通信コントローラへ受信したデータを、ホストへエラー情報を提供します。1 チャンネルに 1 つ装備されます（ネットワーク二重化の場合は 1 ノードに 2 つ装備）。

④バスガーディアン（BG）：

バスドライバによるネットワークへの送信を監視します。通信コントローラ

ラとは別のタイムスケジュールを持っており、常にバスドライバのネットワークへのアクセスをモニタします。そして、不適切なタイミングでの送信要求を検知した場合、バスドライバに対して送信を停止させることができます。1チャンネルに0~1つ装備されます（オプション）。

これらにより、高い安全性、信頼性を実現できます。

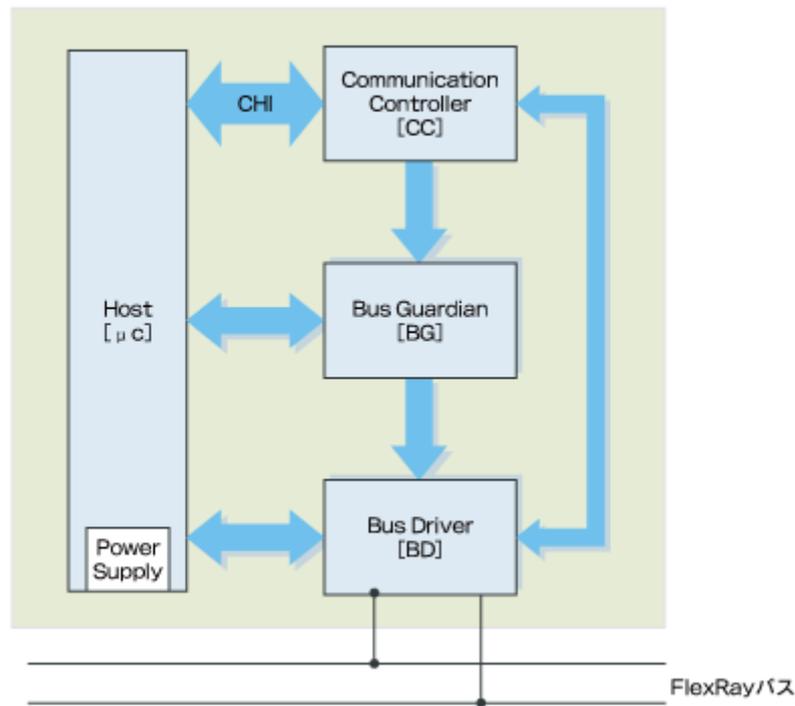


図 118 FlexRay ノード構造：各機能体の相互監視（出典：monoist）

〔5〕優れた通信同期機能

「マルチマスタ・クロック同期」と呼ばれる、任意ノードのクロック（処理のタイミングを合わせるために用いられる信号）に基づく同期方法と、さまざまな補正機能を備えることによって、ネットワーク上のすべてのノードが同じタイミングで通信を行うことができます。

まず通信開始時に、タイミングに関して基準となる複数ノードから「グローバルタイム」が与えられます。このグローバルタイムとは、絶対的な時間ではなく、あくまでも「ネットワーク内における時間に関する共通の“ものさし”（具体的には、サイクルの長さや開始時間）」です。これを用いて各ノードがタイミングを合わせることで同期を図ります。自動車ネットワークの世界時間のようなものです。また通常、温度変化、電圧変動、水晶発振器の製造誤差などにより、各ノードが持つ時間（ローカルタイム）は異なっており、通信開始時

にタイミングを合わせたとしても徐々に差が出てくるため、通信開始後の補正が必要となります。FlexRay では、各ノードが常にグローバルタイムとローカルタイムとの差を測定し、補正することで同期通信を実現しています。

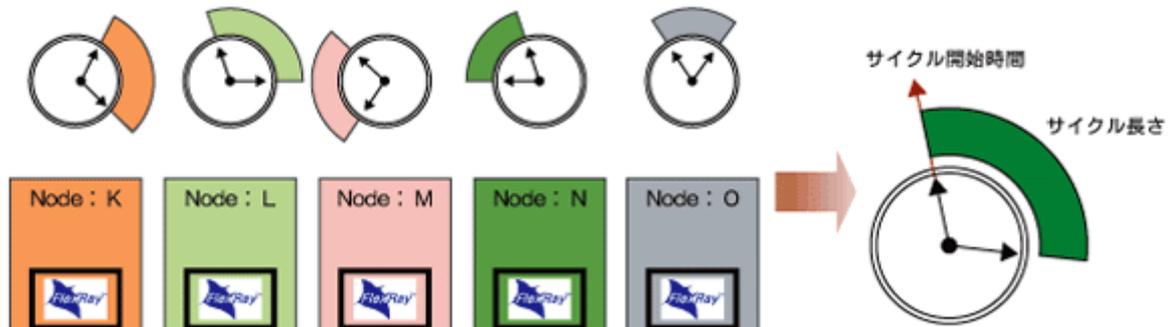


図 119 グローバルタイムとローカルタイム：共通のタイムベースに対してクロック同期を実施（出典：monoist）

〔6〕高速通信

FlexRay の最大通信速度は 10Mbps であり、CAN (1Mbps) の 10 倍の通信速度です。また、前述のネットワーク二重化とも関連しますが、仮に、二重化したネットワークにそれぞれにまったく別のデータを送信した場合、その通信速度は論理的には最大 20Mbps となります

6. 3. 5. 通信構造／バスアクセス

〔1〕サイクル

FlexRay 通信は「サイクル」と呼ばれる、周期を繰り返す“時分割通信方式 (TDMA)”です。このサイクルは FlexRay 通信で基本となる最も大きい単位です。その概要を以下に示します。

サイクルの長さは、ネットワーク全体 (FlexRay では通常「クラスタ」と呼びます) で共通の値として決められます。

1 つのサイクルは、以下 4 つのセグメントで構成されます。各セグメントの長さもクラスタで共通の値として定義されます。

- ① スタティックセグメント (必須)
- ② ダイナミックセグメント (オプション)
- ③ シンボルウィンドウセグメント (オプション)
- ④ NIT (Network Idle Time) セグメント (必須)

各ノードは、サイクルの繰り返す回数を「サイクルカウンタ」として 0 からカウントします。このカウンタ数はある回数でリセットされ、これを「サイクルカウンタワードラップ」といいます。プロトコルバージョン V2.1 におけるサイクルカウンタワードラップは定数=64 と規定されており、サイクルカウンタは 0 を基点に 63 まで増加した後、0 に戻ることを繰り返します (0、1、2、…、62、63、0、1、2、……)。一方、プロトコルバージョン V3.0 以降では 8 ~64 間の各偶数値をサイクルカウンタワードラップとして任意に設定できます。

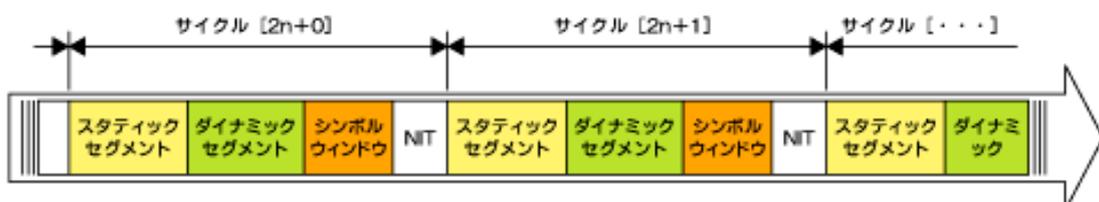


図 120 通信構造—「サイクル」(出典: monoist)

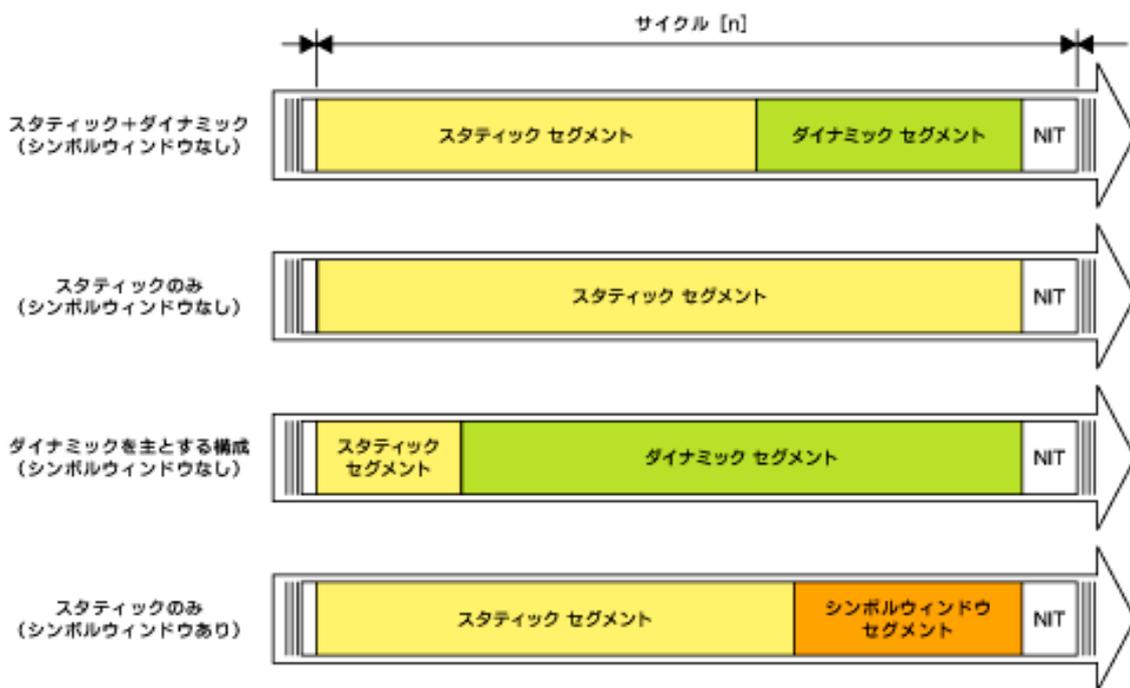


図 121 サイクル構成のバリエーション: 通信構成を柔軟に設計できる (出典: monoist)

①スタティクセグメント

Fixed TDMA 通信を行うための領域です。決まったフレームを決まったスロットタイミングで送信するため、一定の送信周期を保証でき、安全性重視のアプリケーションに適しています。その特徴は、

◇各サイクルの最初に必ず設定されます。

◇「スロット」で分けられます。

(スタティックセグメントのスロットを「スタティックスロット」と呼びます)。1 つのスタティックスロットが経過するたびにスロットカウンターが1 つずつ増えます(1、2、3、……)。

◇各スタティックスロットの長さはクラスタ共通で、すべて同じ長さです。

◇各スタティックスロットに対して送信ノードが割り当てられます。プロトコルバージョン V2.1 では、1 スタティックスロットに1 ノードを割り付けます。一方、プロトコルバージョン V3.0 以降では特定のスロット以外であれば1 つのスタティックスロットに複数のノードを割り当てることが可能です。

◇送信タイミングがあらかじめ決まっているため、バスガーディアンによる保護が可能です。

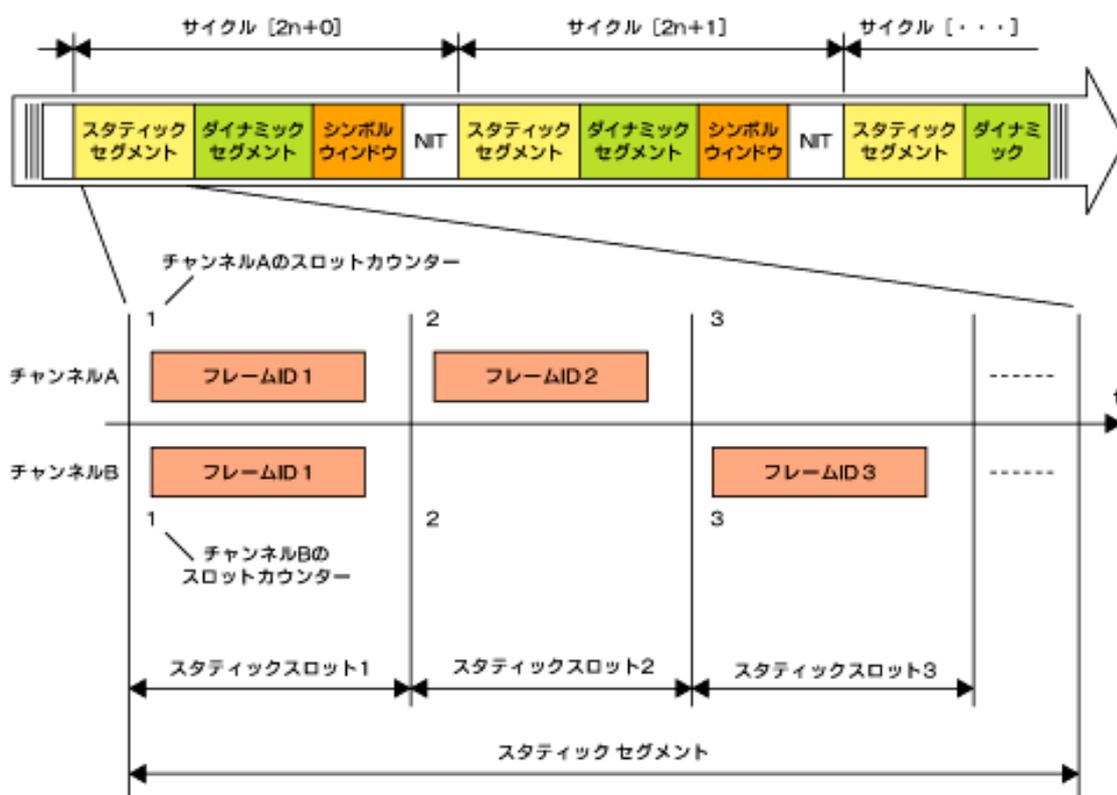


図 122 スタティックセグメントの構造 (出典: monoist)

②ダイナミックセグメント

Flexible TDMA 通信を行う領域です。送信の順番は事前に定義されますが、送信自体は要求に基づいて行われるため、高い応答性が求められるアプリケーションに適しています。

◇スタティックセグメントの次に設定されますが、必須ではありません。

◇「ミニスロット」で細分化されます。1 つのミニスロットが経過するたびにスロットカウンターが1 つずつ増えます(1、2、3、……)。

- ◇送信要求があった場合のみ、複数のミニスロットで構成されるダイナミックスロットとなります。ダイナミックスロットの長さはフレーム長に応じて可変です。
- ◇複数の ECU から同タイミングで送信要求があった際も、事前に定義された送信順に従って送信されるため、フレームの衝突およびそれに伴う送信遅延などは発生しません。
- ◇各チャンネルの各ダイナミックスロットに対して、別々に送信フレームが割り当てられます。
- ◇送信要求がなかった場合、1 ミニスロット分が未使用のまま、次のミニスロットにスロットカウンタが進みます。ミニスロットがいわば「プレイスホルダ（場所取り）」の役割を果たします。
- ◇送信要求に応じて、2 つのチャンネル（A/B チャンネル）はそれぞれ独立してスロットカウンタを進めます。よって、同じタイミングに異なるスロットカウンタ値を取り得ます。
- ◇要求に基づいた送信であるため、バスガーディアンによる保護はありません。

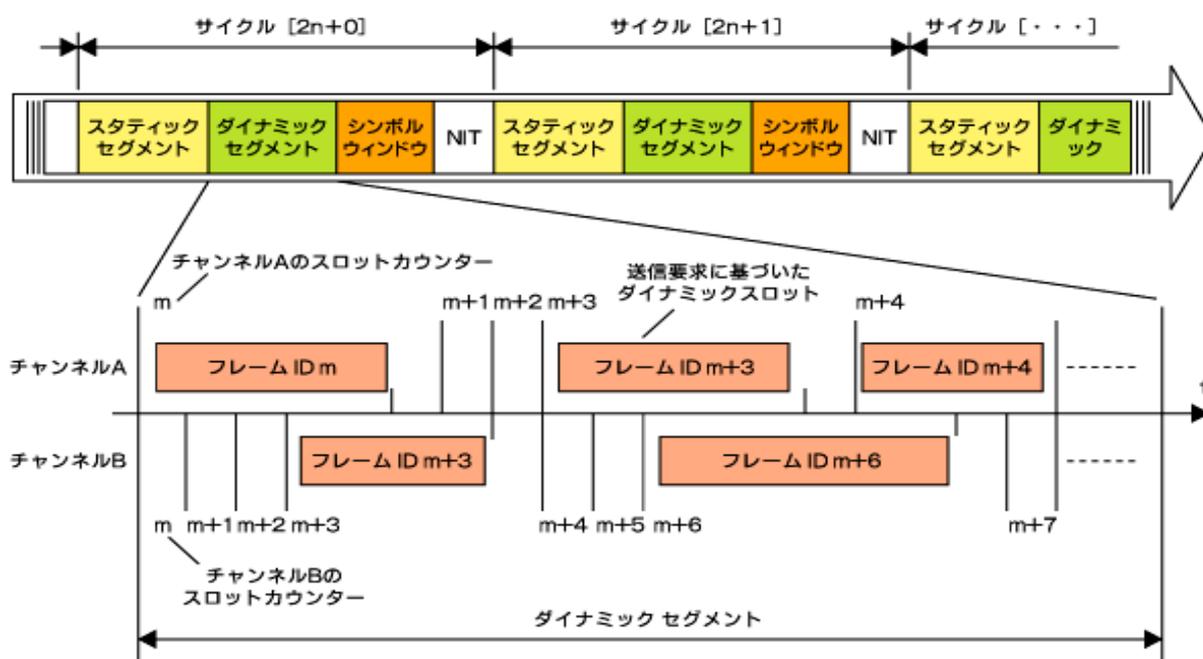


図 123 ダイナミックセグメントの構造（出典：monoist）

③シンボルウィンドウセグメント

ダイナミックセグメントの次に設定されますが、必須ではありません。FlexRay プロトコルにて「シンボル」と呼んでいる特定のビットパターンを送信する領域です。

プロトコルバージョン V2.1 では、送信できるシンボルは「MTS (Media Access Test Symbol)」のみです。この MTS は各ノードのバスガーディアンが正常かどうかをチェックするために使用されます。プロトコルバージョン V3.0 以降では MTS に加えて、「WUDOP (Wakeup During operation Pattern)」と呼ばれるシンボルも送信できます。

④NIT (Network Idle Time) セグメント

各サイクルの最後には、必ず「NIT セグメント」が設定されます。同期に必要な各パラメータの計算を実施する領域、かつ同期のための補正を行うためのいわゆる“のりしろ”となる領域であるため、データの送信などは一切ありません。ネットワーク上ではアイドル時間となります。

⑤サイクルマルチプレキシング

前述のスタティック／ダイナミックセグメントに関する図では、通信例として1つのサイクルのみを示しましたが、各サイクルに異なる通信スケジュールを定義することができます。これによりセグメントを有効に活用できます。これを「サイクルマルチプレキシング」と呼びます。

これは、各フレームに対して“何サイクルに1回送信するか”を示す「サイクルカウンタフィルタ」を設定することにより実現されます。例えば、あるフレームがサイクルカウンタフィルタ=2によってスロット3に設定され、最初のサイクル(=サイクル0)から送信される場合、このフレームが送信されるタイミングはサイクル0、2、4、6、……でのスロット3となります。サイクル1、3、5、7、……でのスロット3には別のフレームを定義できます。

サイクル NO.	スロットNO.														シンボル ウィンドウ	NIT
	スタティック セグメント							ダイナミック セグメント								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14		
0	a	b	c	e		a	f		h	i	j			m		
1	a		d			a			h		k					
2	a	b	c			a	g		h		j	l		m		
3	a		d			a			h		k					
4	a	b	c	e		a	f		h		j			m		
5	a		d			a			h		k					
6	a	b	c			a	g		h		j			m		
7	a		d			a			h		k					
...	a	b	c	e		a	f		h		j			m		
63	a		d			a			h		k					

← 同一スロットを複数フレームで共有

図 124 サイクルマルチプレキシングの例 (出典: monoist)

フレームcがサイクルカウンタフィルタ=2によってスロット3に設定され、最初のサイクル0から送信されるため、その送信タイミングはサイクル0、2、4、6、……となる。同じスロットのサイクル1、3、5、7、……では別フレームが送信できます。

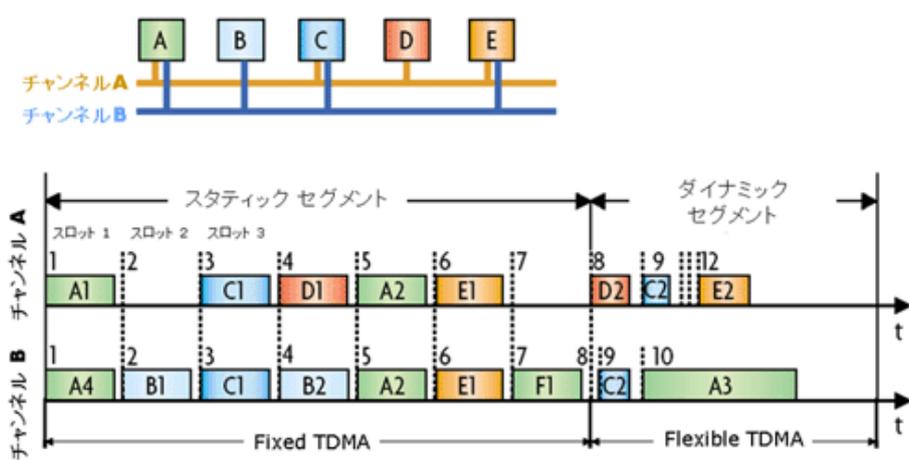


図 125 FlexRay 通信の例 (出典 : monoist)

6. 3. 6. タイミング

FlexRay はタイムトリガー通信方式であり、この方式で重要なのは各ノード共通の“時間軸—タイムベース”です。FlexRay ではクラスタ内の時間における共通の“物差し”である「グローバルタイム」が与えられ、このグローバルタイムと各ノードが持つ固有の「ローカルタイム」の差を測定・補正することで同期通信を実現しています。

これらタイムベースは、先に紹介したサイクルのほかに、「マクロティック (Macrotick)」 「マイクロティック (Microtick)」といった 3 つの“単位”に基づいています。それらの関係は下図のようになります。

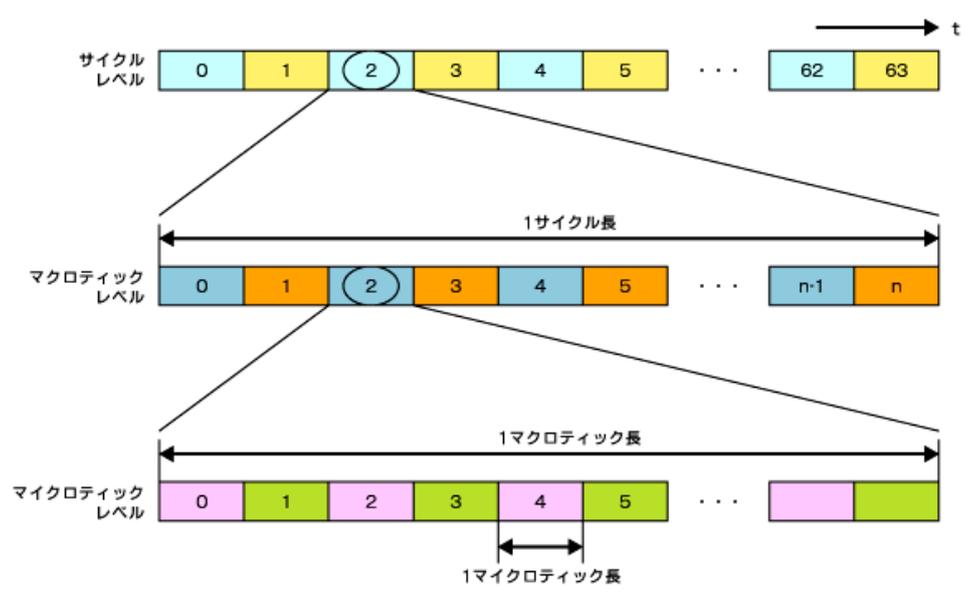


図 126 タイミングの階層 (出典 : monoist)

[1] サイクル

FlexRay 通信における最も大きな単位です。偶数のマクロティックから成り

立っています。

[2] マクロティック (Macrotick)

「グローバルタイム」の最小単位であり、クラスタ全体で共通の値です。複数のマイクロティックから構成されます。

[3] マイクロティック (Microtick)

サンプルクロックから得られる「ローカルタイム」の最小単位です。サンプルクロックは発振器の周波数などノード固有の要因によって決まるため、当然、このマイクロティックの長さも各ノードで異なってきます。

6. 3. 7. フレームフォーマット

フレームは「ヘッダー (Header)」「ペイロード (Payload)」「トレーラー (Trailer)」の3つのセグメントから構成されており、この構造はスタティックセグメントで送信されるフレーム (=スタティックフレーム) とダイナミックセグメントにて送信されるフレーム (=ダイナミックフレーム) で共通です。

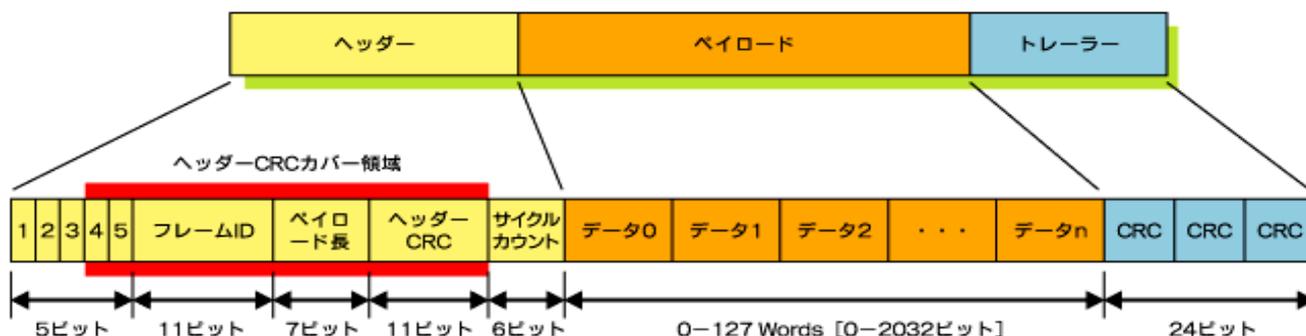


図 127 フレーム構造 (出典: monoist)

それぞれのセグメントに格納されるデータは以下になります。

[1] ヘッダー (40 ビット)

- ◇リザーブ (Reserved) (1 ビット): 今後の拡張のための予約ビットです。通常「0」が設定されます。
- ◇ペイロードプリアンブルインジケータ (Payload Preamble Indicator) (1 ビット): ペイロードの一部を特別な目的に使用するかどうかを示します。
- ◇Null フレームインジケータ (Null Frame Indicator) (1 ビット): フレームが Null フレームかどうかを示します。
- ◇同期フレームインジケータ (Sync Frame Indicator) (1 ビット): フレームが同期フレームかどうかを示します。

ームが同期フレームかどうかを示します。

- ◇スタートアップフレームインジケータ (Startup Frame Indicator) (1 ビット) : フレームがスタートアップフレームかどうかを示します。
- ◇フレーム ID (Identifier) (11 ビット) : フレーム ID を定義します。取り得る値は 1~2047。0 は無効です。
- ◇ペイロード長 (Payload Length) (7 ビット) : ペイロード部分の長さを単位「Word」で示します (※1Word=2バイト)。
- ◇ヘッダーCRC (Header CRC) (11 ビット) : ヘッダー中の同期フレームインジケータ、スタートアップフレームインジケータ、フレーム ID、ペイロード長部分を保護する CRC (Cyclic Redundancy Check/巡回冗長検査) コードです。
- ◇サイクルカウント (Cycle Count) (6 ビット) : サイクルカウンタ数を示します。取り得る値は 0~63 です。

[2] ペイロード (0~2032 ビット)

- ◇データが格納される部分です。最大 127Words (=254 バイト) のデータを定義できます。

[3] トレーラー (24 ビット)

- ◇フレーム CRC (Frame CRC) : ヘッダー全体とペイロードを保護する CRC です。

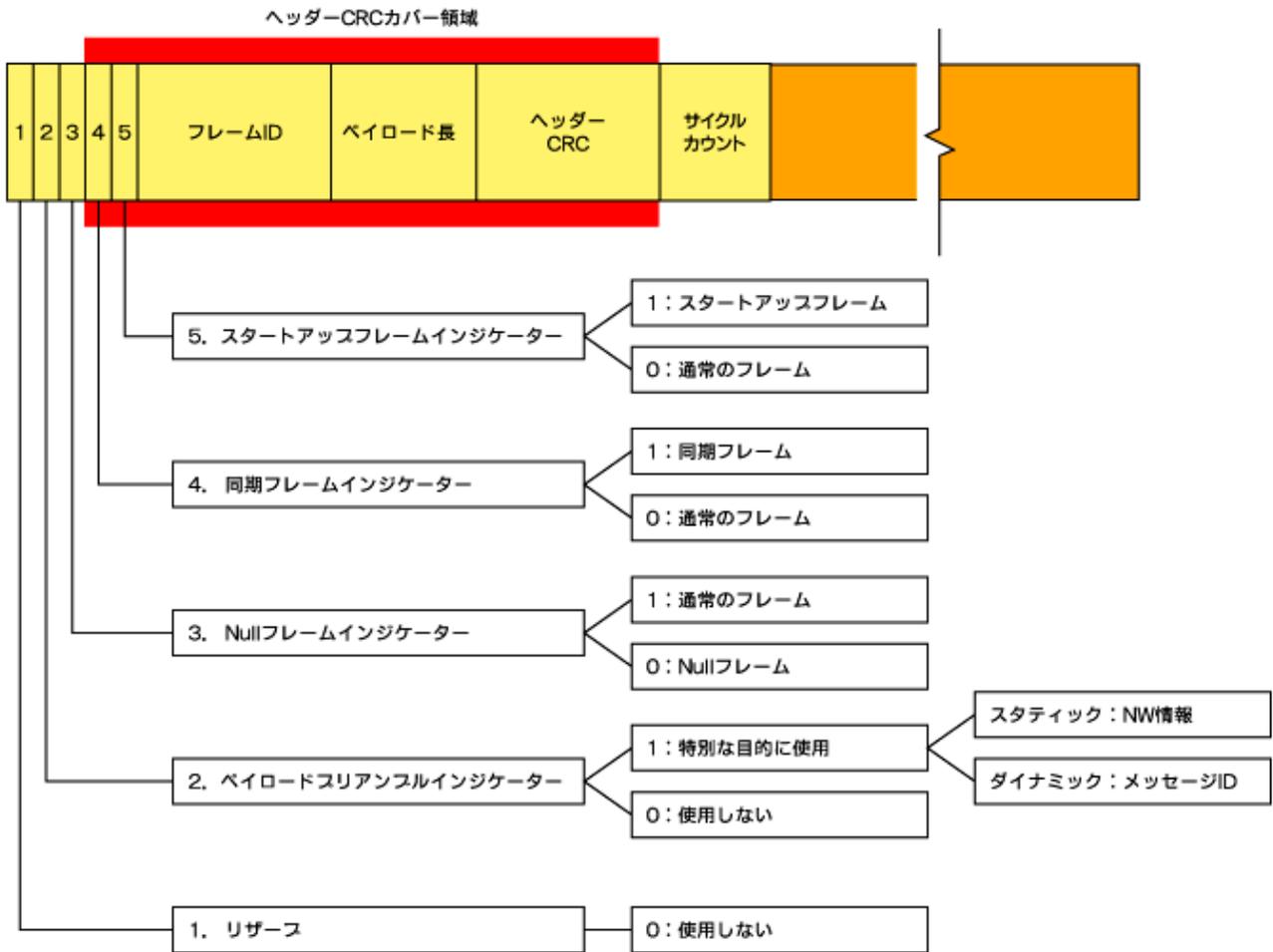


図 128 ヘッダー 5 ビット部分の詳細 (出典: monoist)

[4] ペイロードプリアンブルインジケータ

ペイロードの一部を特別な目的に使用するかどうかを示すビットとなり、その用途はスタティックフレームとダイナミックフレームで異なります。

スタティックフレームではペイロードセグメントの最初の 0~12 バイトにネットワークマネジメントの情報を、ダイナミックフレームではペイロードセグメントの最初の 2 バイトにメッセージ ID を定義できます。

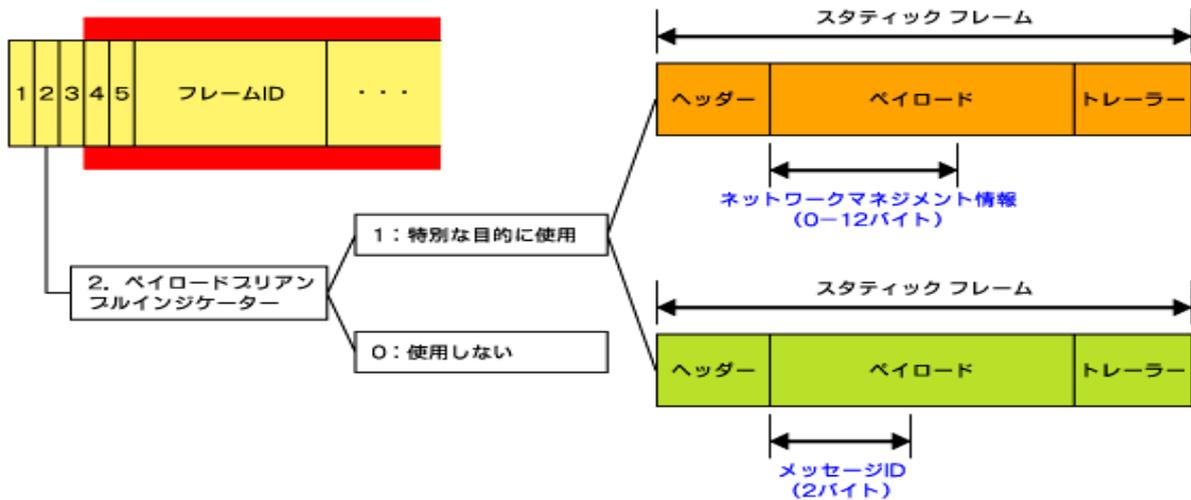


図 129 ペイロードプリアンブルインジケータ (出典 : monoist)

[5] Null フレーム

ペイロードに使用できるデータを載せていないフレーム、つまり“データが使われない”フレームです。フレームのヘッダーセグメントの Null フレームインジケータが「0」のとき、そのフレームは Null フレームとなり、ペイロードデータはすべて「0」となります。このフレームは、例えばサイクルマルチプレキシングの未使用スロットでの送信など、送るべきデータがないとき、またはコントローラでのフィルター条件に合致した場合、スタティックセグメントでのみ送信されます。

[6] 同期フレーム

同期補正を実行する際に使用されるフレームです。

[7] スタートアップフレーム

クラスタ全体の通信開始時に、同期を図るために使用するフレームです。

6章4節 シリアル通信 (RS-232c)

6. 4. 1. シリアル通信とは

パソコン (マイコン) と周辺機器、あるいはパソコン (マイコン) とパソコン (マイコン) 間などで通信を行う技術は様々に存在していますが、最も多用されている通信技術はシリアル通信です。

コンピュータの世界では、扱うデータは全て 2 進数に変換して処理をします。通信を行うためには、この 2 進数のデータをどのようにして送信・受信するのでしょうか。

その方法は、大きくシリアル通信とパラレル通信に分けられます。

シリアル通信は 1 本の信号線を使用して 1 ビットずつ順番にデータを伝送する方法のことを言います。



図 130 シリアル通信

シリアル通信は 1 本の信号線を使用して 1 ビットずつ順番にデータを伝送する方法です。それに対して、パラレル通信は複数の信号線を使用して複数ビットを一度に伝送する方法のことです。

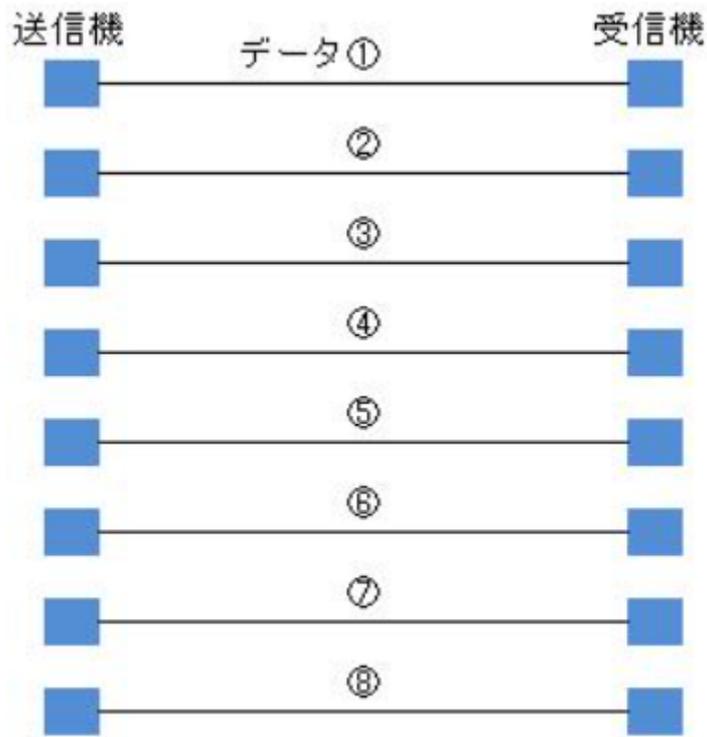


図 131 パラレル通信

パラレル通信を実現するためには何本ものケーブルが必要になり、距離が離れれば離れるほど、コストが膨らんでしまいます。また、ケーブルも太くなってしまうので、きれいにまとめることが難しくなります。通信異常がケーブルに起因して発生したような場合を想定すると、シリアル通信の方が保守に関わるコストも抑えることができるのは、容易に想像できます。

パラレル通信の場合は、一度に複数ビットを伝送するということが送信タイミングのずれを誘発させる可能性があり、通信の精度についても品質が低下してしまいます。

その点、シリアル通信であれば、ケーブルは一本で済むためコストも場所も取らず、通信タイミングのずれも発生しません。

6. 4. 2. RS-232c とは

シリアル通信は、様々な用途に使われるようになりましたが、その際、仕様の互換性を持たせるために、規格統一が必要となり、アメリカの EIA (Electronic Industries Association) が 1969 年にシリアルインターフェースの規格が制定されました。その規格が RS-232c です。

RS-232c では、パソコンなどの DTE（データ端末装置）と、モデムなどの DCE（データ回線終端装置）とを接続してデータ通信を行うための電氣的・機械的な特性が定義されています。（DTE、DCE という用語は RS-232c のなかで定義されたインターフェース用語です。）

6. 4. 3. RS-232c の信号レベル

RS-232c はいくつかの信号線で使用しますが、その中に送信線、受信線、接地線が存在します。

コンピュータで扱うデータは全て 2 進数に変換して処理されますが、この 2 進数のデータを電圧（信号レベル）で表します。信号線が $-3V \sim -25V$ のときが”1”であり、 $+3V \sim +25V$ のときが”0”と取り決められました。また、データを伝送してないときは”1”の状態になるように定められています。

状態	Low	High
電圧範囲	$-3V \sim -25V$	$+3 \sim +25V$
値	”1”	”0”
名称	マーク	スペース

図 132 RS-232c の信号レベル

一見、データ伝送してないときは”0”の状態にする方が正解のように思えるかもしれませんが。これはシリアルインターフェースのルーツがモールス信号にあることに由来します。モールス信号では、データ伝送してないとき電流が流れてなければ、何らかの故障か信号線の切断としていました。その名残が RS-232c 規格に残っているのです。

6. 4. 4. シリアル通信のイメージ

シリアル通信のイメージは、糸電話にたとえることができます。一方は喋る人、もう一方は聴く人です。お互いに話したいときは 2 本の糸電話が必要です。ただし、同時に使用することはできません。

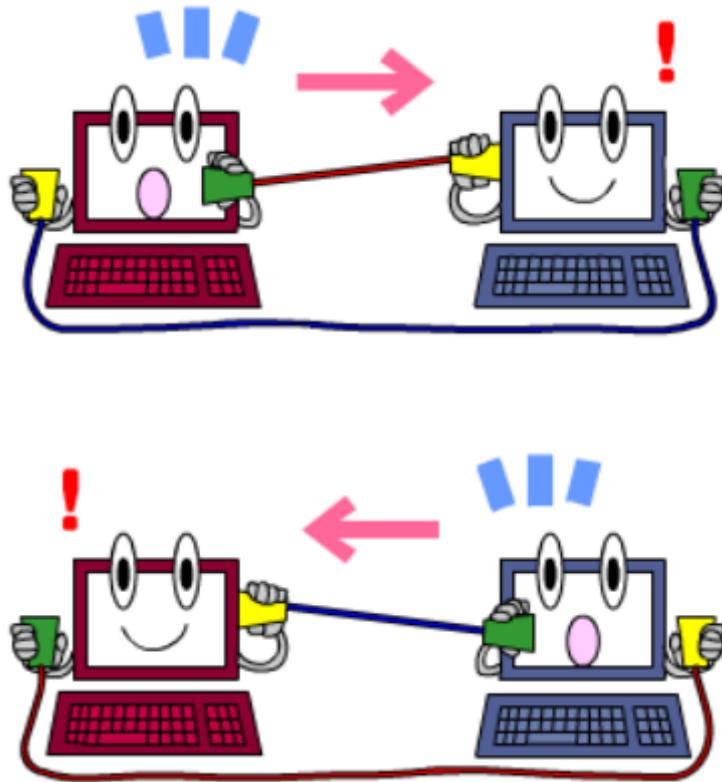


図 133 シリアル通信のイメージ (出典 : CodeZine)

一方から「お元気ですか」と送ると、返事として「元気ですよ」と帰ってきます。一本の糸電話で「お元気ですか」の文字列が送られます。この通信方法がシリアル通信です。パソコンの場合、文字は1文字(1バイト)ずつ送ります。

相手に受け取り準備をしてもらうため、「送りますよ」「1文字目」「終わりました」「送りますよ」「2文字目」「終わりました」…と1文字ごと、送り初めと終わりを知らせます。送る文字の内容は送る側と受け取る側で独自に決めることができます。

こちらから送りたい相手への線は1本、相手からこちらへの線1本、それに接地線の3本で通信できるということになります。

この3本線が一つのケーブルに入ってシリアルケーブルができあがっています。

6. 4. 5. 信号の種類とコネクタ

RS-232cは、使用するコネクタや信号配置も規定され標準化されています。図にD-SUB9ピンの信号配置と信号線の種類を示します。

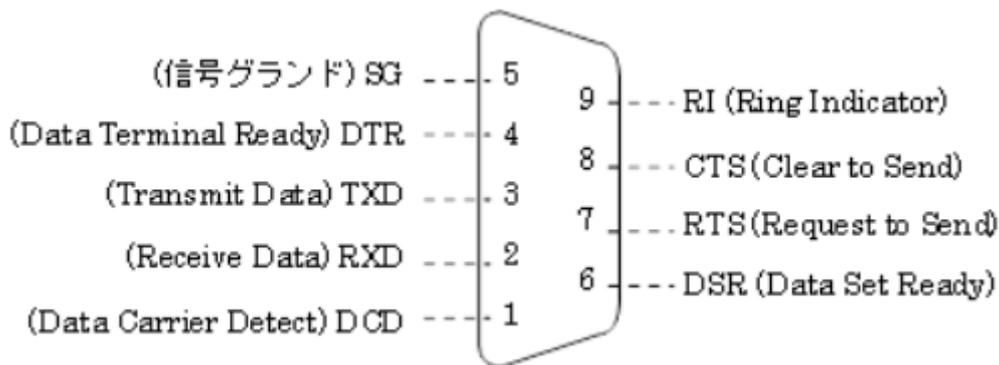


図 134 信号線の配置 (D-SUB9 コネクター)

ピン番号	信号名	説明	
1	DCD	Date Carrier Detect	キャリア検出
2	RxD	Recieved Data	受信データ
3	TxD	Transmitted Data	送信データ
4	DTR	Data Terminal Ready	データ端末レディ
5	SG	Signal Ground	信号用接地または共通帰線
6	DSR	Data Set Ready	データセットレディ
7	RTS	Request To Send	送信要求
8	CTS	Clear To Send	送信許可
9	RI	Ring Indicator	被呼表示
CASE	FG	Frame Ground	保守用接地またはアース

図 135 信号の種類 (出典 : CONTEC)

上の図は、D-SUB9 ピンコネクターの信号線ですが、D-SUB25 ピンコネクターも存在します。現在では、パソコンにシリアルポートのコネクターが実装されていないものが多い、USB を仮想シリアルポートとして利用する「USB-Serial 変換ケーブル」というものが市販されています。この「USB-Serial 変換ケーブル」では、USB の規格に従って、データの送受信を行う信号は、D+と D-の 2 本しかありません。(線間差動信号と云います。) このために、上の図で示す RxD、TxD と接地信号以外は、使用されません。

シリアル通信の接続対象にもよりますが、糸電話で説明したように、シリアル通信では通信用の信号と接地信号の 3 本の信号があれば、通信ができるので、いろいろなところに使用されてきたわけです。

6. 4. 6. 通信手順と実際の接続

下の図は、2 台の PC がシリアル通信を行う手順です。

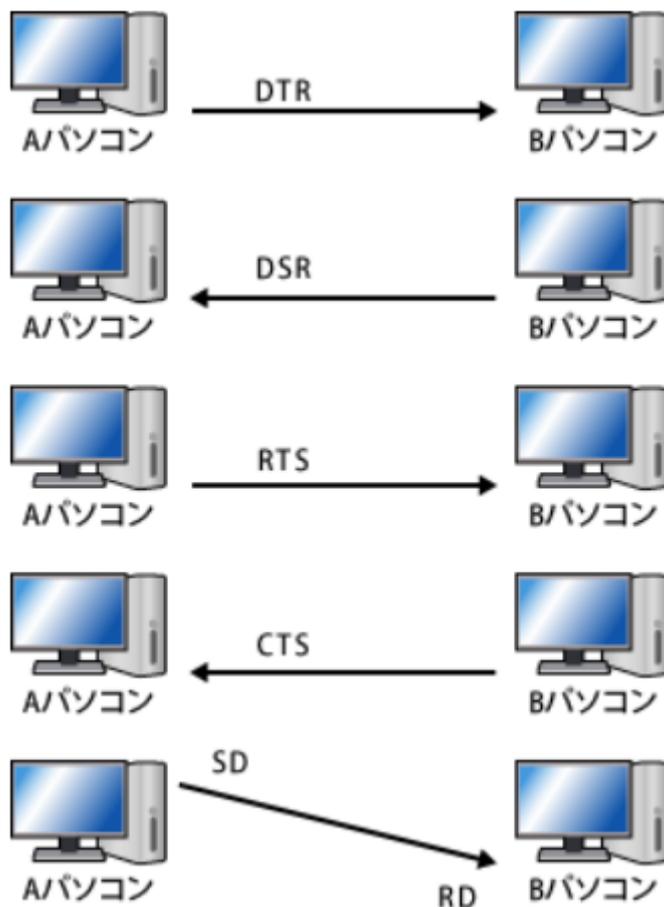


図 136 正式な通信手順 (出典 : CodeZine)

- ① まず A パソコンから DTR (信号を送る通路が使えます) を ON にします。B パソコンは DSR (私も準備 OK です) を ON にします。
- ② 次に A パソコンは RTS (データを送ってもよいですか) を ON にします。B パソコンは CTS (送ってもらって大丈夫です) を ON にします。
- ③ これで準備が整いました。A パソコンは TxD から信号を送ります。B パソコンは RxD から信号を受け取ります。

以上のような手順を行いながら、対向機 (通信相手) との接続信号数を減少させられる接続方法が次の図です。

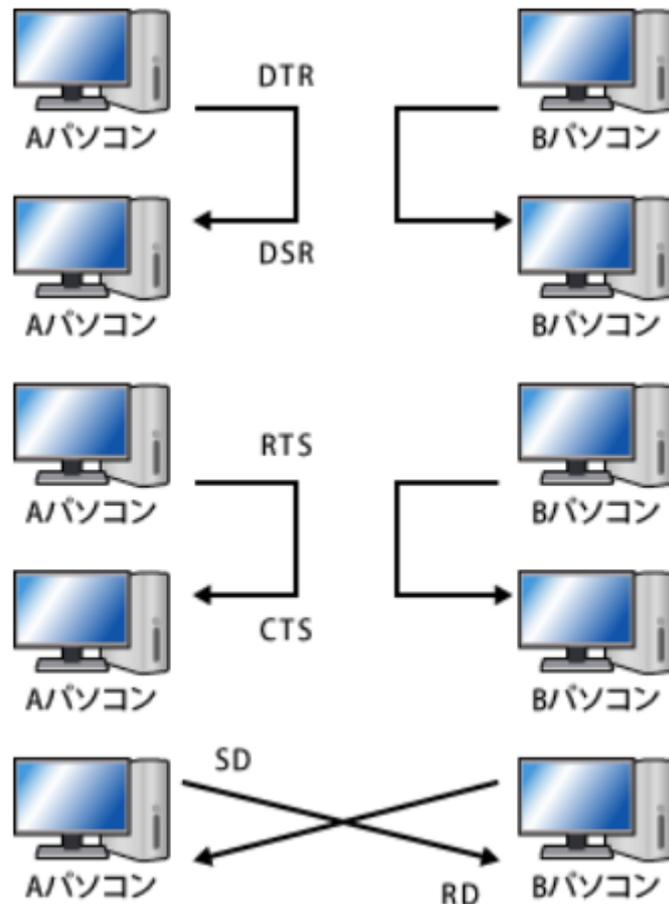


図 137 簡易な接続 (出典 : CodeZine)

もし A パソコンが DTR を出して DSR 待ちの時、自分が出した信号線がそのまま DSR につながっていると A パソコンは DSR 信号を相手から貰ったと思い次のステップへ行きます。RTS も同様に自分が出した信号をそのまま CTS につなぐと送信可能と思い次のステップへ行きます。相手からの通信待ちで止まってしまうことなく接続開始ができるようになります。

そこで、ケーブルを作る時、4:DTR と 6:DSR のピンを自分のコネクタの中でつなぐのです。7:RTS と 8:CTS もつなぎます。これで接続開始の処理は自分のパソコン側のみでできるようになりました。そうすると実際に必要な信号は 2:RxD、3:TxD、5:GND の 3 本になります。

A パソコンの 2:RxD と B パソコンの 3:TxD をつなぎ、A パソコンの 3:TxD と B パソコンの 2:RxD をつなぐので信号線をクロスさせることからクロスケーブルと言います。

6. 4. 7. 信号線の中のデータ

送受信信号の中を伝わるデータはどのようなものでしょうか。

図に1バイトのデータが送られる様子を示します。

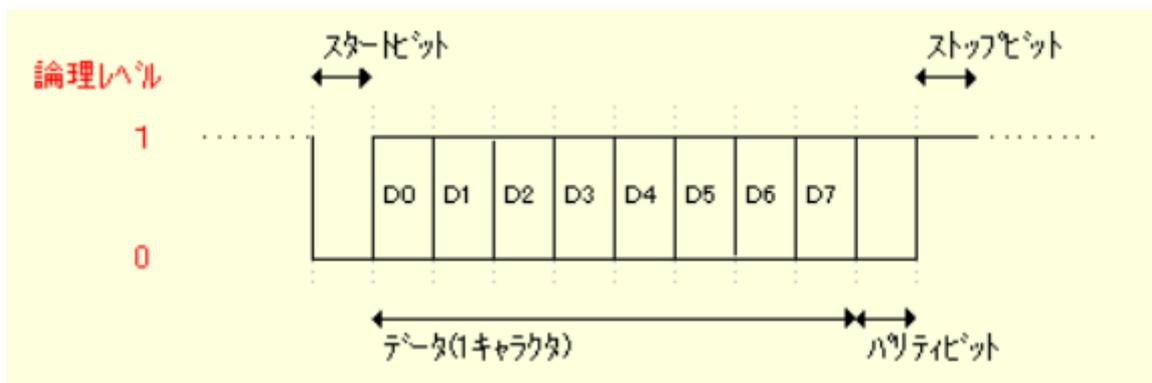


図 138 1バイトの送信データ (出典：Softech)

図では、左側に向かってデータが送られる様子を示しています。

インターフェース IC の送信レジスタに1バイトのデータを書き込むと、最初にスタートビットが送られて、相手に「これから通信データが来る」ことを知らせます。続いて、8bitのデータが送信され、後に誤り検出のためのパリティビットが続きます。最後にストップビットが送られて1バイトのデータの送信が終わります。

スタートビットは、1ビットですが、ストップビットはパリティビットの有無などにより、1~2ビットになります。パリティビットは偶数パリティ・奇数パリティ・パリティ無しが設定できます。パリティ無しに設定した場合は、ストップビットを2bitにします。

シリアルインターフェースのメーカーや使用する IC の種類によっても、設定の内容が異なるので、直接このような設定を行うときは、IC のデータシートを良く読むことが必要です。

様々な家電品や制御機器などに内蔵されるシリアル通信部分を開発する場合は、このような制御設定を細かく行うことにより、通信精度・通信品質を高めます。

通信速度はボーレート (Baud Rate) と呼び、単位は bps (Bits Per Second) で表します。

1秒間に送信できるビット数を意味します。このボーレートは、スタートビット・データ・パリティビット・ストップビットのすべてを含めた送信速度なので、データ自体の通信速度は、ボーレートより遅くなることに注意してください。

D0	D1	D2	D3	D4	D5	D6	D7	P
1	0	0	0	0	0	1	0	0

図 139 'A'のデータ (偶数パリティ)

上図に半角の A (16 進数の H'41) を偶数パリティで送信する場合を示します。この時の実際の通信信号をロジックアナライザで見ると、次のようになります。

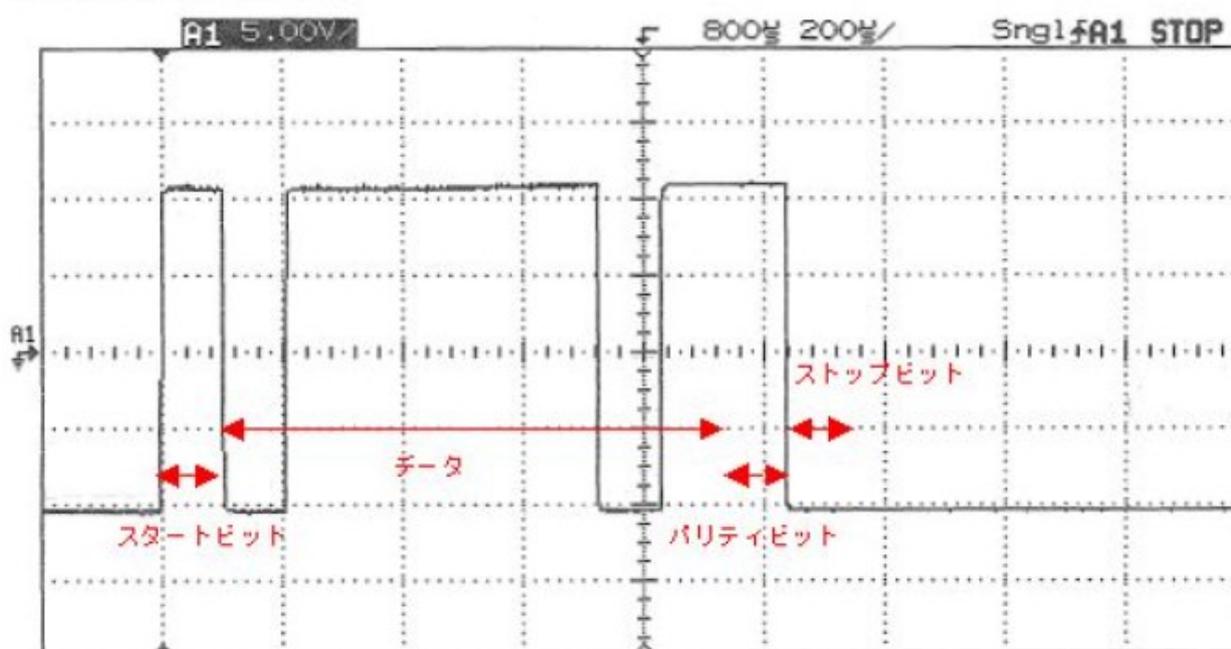


図 140 実際の送信信号 (9600bps H'41='A'・1ストップビット・偶数パリティ)

組み込みソフト開発ではソフトウェアの開発とともにハードウェアの開発することが良くあるため、ハードウェア側に不具合が存在する可能性もあります。実際の波形を見てどのように通信が行なわれているかを理解することができれば、いろいろな側面からの問題解決が出来るようになります。

6. 4. 8. シリアル通信の実験

9章の最終節でシリアル通信を利用した実験を行っています。この節で説明したシリアル通信では、「いざプログラム」となると、少し難しく思われたのではないのでしょうか。しかし、実際には様々なライブラリが提供されている環境を利用すれば、いとも簡単にシリアル通信を利用することができます。その良い例が9章の最終節です。是非参照して、実験してください。

7章 自動車と制御

地球温暖化問題や都市環境問題を解決するために、エネルギー利用効率が優れ、CO₂ 排出量の少ないクリーンな自動車の開発がすすめられています。エンジンとモーター動力を効率よく使い分けて、制動や減速時に捨てていた機械的/熱的エネルギーを回生し電力として再利用する、優れた燃費性能を持つハイブリッド車が、1997年に実用車両として市販されました。また将来の燃料電池車につながる技術として応用開発が進められています。この章では、現在の主流であるハイブリッド車を中心に自動車制御技術を概観します。

7章1節 電気自動車の制御技術体系

表7に電気自動車の技術体系を示します。

将来的な電気自動車の姿として、水素燃料電池自動車が期待されていますが、燃料電池自体の技術開発や燃料インフラの整備という点では、まだ多くの技術的・社会的課題が残っています。燃料電池の章で説明した、ガソリン等を「改質」して水素を含むガスを発生させ、それを利用する改質器を搭載した燃料電池車についても開発が進められていますが、高度な技術開発が必要とされ、実用化には時間がかかる見込みです。資源エネルギー庁長官の私的研究会「燃料電池実用化戦略研究会」では、2010年に5万台を導入する目標を掲げていましたが、達成されていません。本格的な量産はまだ10年以上必要との意見もあります。これらのことから、当分の間はガソリンを中心とした液体燃料を利用する内燃機関（エンジン）と電気モーターを組み合わせたハイブリッド電気自動車が主力として商品化が進められるものと思われます。

モーターで車輪を駆動する自動車の共通要素である、モーターやインバータの電力制御、バッテリーなどの電力蓄積デバイス制御、動力源であるエンジンとの最適組み合わせ制御（協調制御）を中枢制御技術としていることが表7からも読み取れます。この技術は燃料電池車のような次世代型自動車の制御技術にもつながる重要分野です。したがって、ハイブリッド電気自動車の制御技術を理解することが電気自動車全体の中枢制御技術を総合的に理解することにつながると考えています。

表 7 電気自動車の技術体系（出典：特許流通報告）

電気自動車 の種類 機能	純電気自動車	ハイブリッド自動車		燃料電池車	太陽電池車
		電気式	蓄圧式		
燃料/ 動力変換	・無し	・有り		・有り	・有り
	・充電→バッテリー蓄積	・液体燃料→内燃エンジン(ガスタービンエンジン含)による動力変換		・水素系燃料(改質含)→燃料電池発電	・太陽エネルギー→ソーラー電池発電
	・急速充電	・燃費効率に優れたエンジン		・小型軽量高性能化、メテフリー電池ユニット	・小型軽量高性能電池ユニット
	・コンタクトタイプ/インダクティブ充電	・ウォーフ、起動性能、エミッション性能に優れたエンジン			
電力蓄積	<ul style="list-style-type: none"> 各種使用法に適合した高性能で低価格なバッテリー 密閉型鉛酸電池 → ニッケル水素電池ユニット、リチウムイオン電池 → 電気二重層キャパシタコンデンサモジュール等の軽量小型でエネルギー密度、出力制御性、メンテナンス性に優れた電池 ユニットからセル単位までの電池残留電力管理制御システム 				
動力/ 電力 変換・ 逆変換	車載 発電機 (レンジエクステンダー)	・エンジン発電機	・油圧モーター (大型車両のみ)	・燃料電池による 直接発電	・太陽電池による 直接発電
	電動 モーター	・高性能モーター、発電兼用モーター → インバーター交流誘導モーター、永久磁石式直流同期モーター・発電機、リラクタン্সモーター等		・高性能モーター、発電兼用モーター → インバーター交流誘導モーター、永久磁石式直流同期モーター・発電機、リラクタン্সモーター等	
減速時 エネルギー 回収	<ul style="list-style-type: none"> 回生制動発電(発電、充電効率の向上) 電気-機械複合ブレーキ 回生ブレーキ用キャパシター(特に電気自動車、大型ハイブリッド車) 		・ポンプ-アキュムレータによる油圧回生	<ul style="list-style-type: none"> 回生制動発電(発電、充電効率の向上) 電気-機械複合ブレーキ 回生ブレーキ用キャパシター(特に電気自動車、大型ハイブリッド車) 	
変速機能 (パワートレイン)	<ul style="list-style-type: none"> 一軸、二軸、ホイールハブ・ホイール等、モーター変速-差動駆動の各種パワートレイン方式がある 電気式 	<ul style="list-style-type: none"> 電気式/機械式およびその組合せ等各種方式 	<ul style="list-style-type: none"> エンジン駆動軸とは独立に油圧モーター・ポンプ系を配置 発進・加速時のトルクアシスト 	<ul style="list-style-type: none"> 一軸、二軸、ホイールハブ・ホイール等、モーター変速-差動駆動の各種パワートレイン方式がある 電気式 	
協調および 全体システム 制御	・モーター、バッテリー出力の協調制御	<ul style="list-style-type: none"> エンジン、モーター、発電、バッテリー出力の協調制御(モーターアシスト式) アイドルストップ・スタート 	・トルクアシスト最適化システム	・燃料電池、モーター、バッテリー出力の協調制御	・太陽電池、モーター、バッテリー出力の協調制御

ハイブリッド電気自動車の制御技術は大きく二つに分けられます。

一つ目はシステム全体の制御です。燃費性能や排気性能の向上のためにシステム全体で最適化を図るために重要な部分で、システム全体が目標に沿うようにマネジメントします。特に、エンジンとモーターなど複数の動力源の間の車輪駆動トルク配分・電力分担などのエネルギーマネジメント、バッテリー等の電力蓄積デバイス間のマネジメント、動力源切替時のショックやアクセル応答性能向上など協調制御が重要な事項となります。

これらは、既存のエンジン制御技術に加え、バッテリー等の電力蓄積デバイスの状況が常に変動する状況でモーターや動力の合成・配分機構を同時並行して制御するために、非常に広範囲で複雑、高度な制御技術です。

二つ目はエンジンやモーター、電力蓄積デバイス、発電機、駆動力の合成・変則機構などの制御です。ハイブリッド電気自動車では、アイドルストップを行うことも多く、温度低下に伴う触媒の能力低下を補うための技術が開発されています。バッテリーはその温度管理や充放電状態により寿命が大きく変化するために、バッテリー単体での制御が必要です。

その他にもエアコンやブレーキなどの車両補機との協調制御等々、非常に多岐にわたります。

7章2節 ハイブリッド電気自動車特有の制御

従来の自動車では、減速時にブレーキにより運動エネルギーを熱に変えて空気中に放出していました。この捨ててしまっていたエネルギーを回収して再利用しようというのがエネルギー回生制御と呼ばれる技術です。減速時には駆動用モーターを発電機として使い、蓄電デバイスに電力として貯蔵し、再加速時にモーターに電力を供給してエンジン負荷を減らしながら従来と同様かまたはモーター出力分を加えた加速性能を実現します。

エネルギー回生制御では、バッテリーの寿命に影響があるため、満充電の状態になると充電をカットします。そのためエネルギー回生を中止する制御を行いますが、この時運転者が違和感を覚えないようにするため、車両の減速状況に変化が現れないようにする必要があり、ブレーキシステムとの協調制御が必要になります。

一方、エンジンと車輪が力学的に接続されていると、エンジンブレーキ作用により、エンジン自身がブレーキとして働くために、回生できるエネルギー量が減少してしまいます。少しでもこの回生量を増やすためにエネルギー回生時はエンジンが機械的に切り離されます。切り離されたエンジンはガソリンの消費量を抑えるために停止するように制御されます。エンジンと蓄えたエネルギーを再利用するモーターの両方による加速では、一般的にモーターの効率がエンジンより勝っているので、できるだけモーターを使用するように制御されます。アイドルストップ機能を持つ場合、モーターのみで発進・加速します。この際、違和感のないエンジンの再始動と切換制御が必要です。

システムマネジメントとしてエンジンを短時間で確実に始動させ、エンジン再始動時のエネルギーを削減して連続性のあるスムーズな加速を実現するために、クラッチ等の切換方法やタイミングを最適制御することと、低温・劣化等で状態が変化するバッテリーに合わせた電力や駆動トルクの管理・制御が重要な要素です。特に駆動力の合成・変速機構では発電機やモーターの速い応答性能を生かした慣性力補償技術等のきめ細かな制御技術が自動車メーカーそれぞれで、数多く開発されています。

排気性能向上については、エンジン停止中に触媒の温度が下がり、触媒性能が低下してしまうことを防ぐために、触媒を予熱して排気性能の低下を防止するように制御されています。

停車中にアイドル停止制御を行って燃費性能を向上させていますが、一方ではアイドルリング中もエアコンや発電機などの様々な補機を駆動しているので、これらの機能をどのようにして、維持するかが技術的な課題です。停車中でも、エアコンのスイッチが入るとエンジンを始動したり、停止中は送風機能に切り換える等の制御が行われています。

その他、灯火類の電力制御や、エンジン停止中でもステアリング操作が可能な電動パワーステアリング制御など、対応している制御技術は多様です。

このような、大規模、複雑なシステムを搭載することで、車両の状況がどのようになっているかを自己診断し、それを運転者に伝える状況表示等との連携制御も必要です。

7章3節 システム全体の制御

7. 3. 1. システムマネジメント制御

燃費性能の向上は、ハイブリッド電気自動車の中心課題です。電力を発生することのできる要素はバッテリー、発電機、回生時の駆動用モーターがあり、一方電力を吸収・消費する要素はバッテリー、走行時のモーターがあります。システムの高効率運転を実現するためにこれらの要素間の電力収支や運転状態を制御する電力マネジメント制御、車両の減速エネルギーを効率よく改正するためのモーターを主としたトルクマネジメント制御が導入されています。

7. 3. 2. 協調制御

駆動源であるエンジンとモーターの動力源切換・合成制御はエンジンとモーターの出力特性の違いを超える高度な制御が要求される、ハイブリッド電気自動車を支える重要な制御です。発進時や加速時に運転者に違和感を与えないように、主にモーターに早い応答特性を利用した切換ショック低減制御や切換タイミング・条件を設定したり、応答性を改善する制御が実用化されています。

7. 3. 3. その他のシステム制御

エネルギー回生時の急制動時のモーター制御や複数のバッテリー・電力蓄積デバイスを搭載してシステムの要求により各々の電力収支を制御する技術が開発されています。

7章4節 要素制御

7. 4. 1. エンジン制御

エンジン制御分野では、システムの高効率化を実現するために、ハイブリッド電気自動車に特化したエンジン運転領域、応答性を前提とした燃料噴射量制御・燃料カット制御や点火時期制御が実装されています。アイドルストップ制御を行う場合は、エンジン再始動制御や既存の可変気筒数制御やバルブタイミング制御との組合せが行われています。

7. 4. 2. モーター制御

モーター制御分野では、エネルギー回生のために走行条件やシステムの状態に応じたモータートルク制御が行われます。加速から一定速度走行へのスムーズな移行制御、エンジンのトルク変動を効率よく補正する制御、複数のモーターで効率を上げる制御、前輪をエンジンで駆動しながら後輪で回生しシステム効率を向上させるような制御技術がすでに実用化されています。

7. 4. 3. 走行性能制御

アクセル応答性や加速度性能を高めてレスポンスの良いスムーズな加速を実現するために、エンジンに比較してトルク応答の速いモーターを利用して過渡的にモータートルクをエンジントルクに加えるトルク制御や、短時間であればモータートルクを定格以上に出せる特徴を生かした、短時間駆動パワーアシスト制御、駆動源切換時の慣性質量によるショックを緩和するようなトルク制御が実装されています。

7. 4. 4. バッテリー制御

バッテリーの劣化検出と補償は、バッテリー寿命に与える影響が大きいため、この分野はハイブリッド電気自動車の重要な制御対象となっていて、バッテリーの劣化に応じた充電シーケンスや充電パラメータの変化を制御したり、放電値の最低値を確保するための放電制御などの充電マネジメント技術が実用化されています。バッテリーが蓄積しているエネルギー量を推定して寿命延長を図ったり、エンジン始動のための電力を確保する制御などが実際に行われています。

7. 4. 5. 動力合成機構制御

エンジンとモーターの努力を合成する機構では、クラッチやモーターを制御してショックを抑えながら切換・合成を素早く行うことが燃費性能・走行性改善につながります。電磁クラッチの電流を周辺状況に応じて細かく制御する切換タイミング制御、水温や油温等停止中のエンジン状況の変化に応じた切換ショック低減制御、回生制動時にクラッチ操作によって、エンジン出力軸と動力伝達軸とを切り離す際のエンジnbrブレーキ分のトルク補正を行うことによる、ショック低減制御などが実装されています。

7章5節 既存制御技術の理解

既が開発されて、実車に搭載されているエンジン燃焼制御システムについて、ホンダが開発した制御システムを紹介します。この節の解説を理解することにより、自動車の制御の基礎が理解できると思います。

7. 5. 1. 燃料噴射制御システム

図は4サイクルエンジンの工程です。第一工程の吸入時に気化した燃料と空気の混合気がシリンダ内に吸い込まれて、後の工程で圧縮・点火・爆発・・・と進んでいきます。

吸気バルブはカムにより開閉しますので、エンジンが低速でも高速でも、同じタイミングで吸気が行われますが、高速運転した時の吸気は、バルブまでの経路により抵抗を受けて、スムーズかつ必要十分な混合気が吸入できない場合があります。これを、コンピュータによる電子制御化して、吸気バルブの代わりに燃料噴射装置を設けて、ピストンの位置やその時の回転数、速度、加速時か減速時か一定速度か、・・・など、諸条件に応じたタイミングと噴射量を制御する方式が、燃料噴射制御システムです。

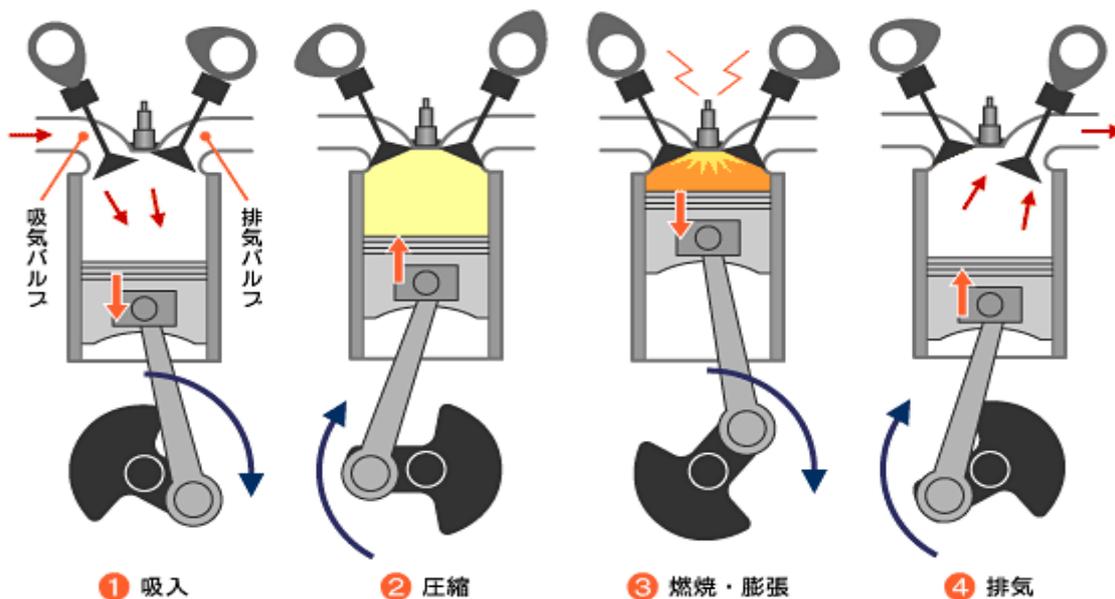


図 141 4サイクルエンジンの工程 (出典：monoist)

燃料噴射制御の要点は基本的に次の2つです。

- ① 燃料噴射量の決定：周囲の状況を各種センサーから読み取り、適切な噴射量を決定します。
- ② 噴射時期の決定：クランクセンサー (※) と上死点センサー (※) により、噴射タイミングを検出します。(※：「7. 5. 3. 噴射時期の決定」を参照してください。)

7. 5. 2. 燃料噴射量の決定

このシステムはエンジン回転数とインテーク・マニホールド（内燃機関の燃焼室に空気を導入するための多岐管）負圧から ECU が基本噴射量を決定するスピード・デンシティ方式を採用しています。ECU はこの基本噴射量をベースに各センサー信号による補正を加えて状況に応じた最適な燃料噴射量を決定して、フューエル・インジェクタ（燃料噴射装置）を駆動します。フューエル・インジェクタは一定ストロークで作動するので、燃料噴射量はインジェクタへ通電する時間の長さで決定されます。

次の図に燃料噴射量決定の流れと関連するセンサーを示します。



図 142 燃料噴射量の決定（出典：科学技術振興機構）

7. 5. 3. 噴射時期の決定

クランクセンサーとカム角センサー（上死点センサー）により気筒判別と各気筒への燃料噴射時期を検出し、フューエル・インジェクタにより各気筒へ 噴射します。

クランクセンサー信号は、クランクシャフトパルサーローターに取付けられたクランクセンサーにより検出されています。

また、上死点（TDC：Top Dead Center）信号は、気筒判別信号カムシャフトパルサーローターに取付けられたカム角センサー（上死点センサー）により検出されています。



図 143 エンジンとセンサー（出典：科学技術振興機構）

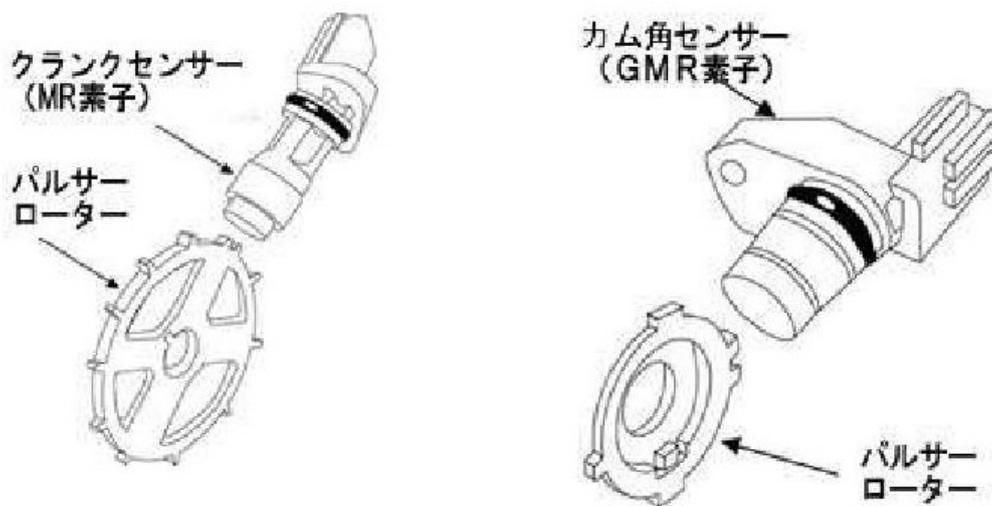


図 144 パルサーローター（出典：科学技術振興機構）

各センサーは、回転に合わせてパルス 信号を ECU に送っています。

ECU は 2 つのセンサー情報から、点火時期、燃料 噴射時期、噴射量を算出しています。

【クランクセンサーとカム角センサーの両方から情報を得る必要性】

① 4 サイクルエンジンでクランクが 2 回廻る間にカムが 1 回廻ります。クランク角から情報だけでたとえば ピストンが上昇しているときに、それがどの気筒圧縮工程なのか排気工程なのかがわからないので、カム軸カム角センサーでそれを判別する必要があり、これを気筒判別と言います。

② 一方でクランク角の方が、分解能が 2 倍高く、またカム角センサーは出力タイミングベルトの影響を受けるため、時期を制御するクランク角センサーの方がより高精度です。

このため、2 つセンサーから情報を得て、点火時期、燃料噴射時期、噴射量を算出しています。

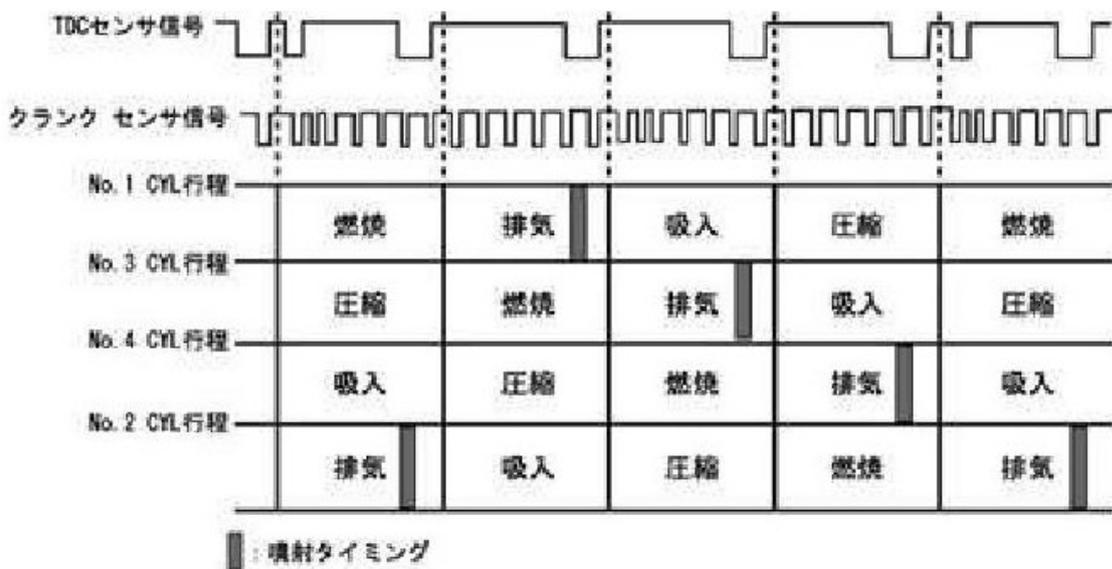


図 145 燃料噴射タイミング (出典：科学技術振興機構)

上図に示すように、TDC センサーとクランクセンサー信号により、気筒判別を行い、噴射タイミングを検出して燃料を噴射します。

7. 5. 4. 燃料噴射制御の難しさ

例えば 6000rpm でエンジンが動いていたとすると、2 回転に要する時間は 20ms となります。20ms の一部の時間で、7.5.2 で図示したいろいろなセンサーの値を読み取り、それらを基に燃料噴射量の決定とタイミングの検出、燃料噴射を行う必要があります。

回転数が上昇すれば、さらに短い時間でこれらの演算処理と制御を行うことが必要になりますので、**制御システムは時間との戦い**になります。また、協調制御のためや、診断のために**通信も発生**します。さらに処理に使える時間が少なくなってしまう。このため、エンジンの制御に用いられる ECU では汎用の CPU を用いられることは無く、専用に開発された CPU が処理を行います。新しいエンジンを開発する際には、エンジンそのものもさることながら、実際の制御を司る ECU を開発しエンジンとのマッチングを図ることが大変大きな仕事となります。

7. 5. 5. マイコンプログラム上のエンジン制御の核心

前述のとおり、エンジン制御は時間との戦いですが、この条件を除外すれば、とても単純です。

- ① センサーで周辺環境を調べる
- ② 対応する制御量を計算する
- ③ 制御量に応じた値をタイミングよくデジタル信号として出力する

①②③をひたすら繰り返すこととなります。

センサーで周辺状況を調べるとは、温度や湿度などを測ることと同じです。また、対応する制御量は、最低 1bit (ON/OFF) です。「暑ければ窓を開ける、寒ければ閉める。」みたいなものです。デジタル信号を出力するのは、プログラムで書けばどんな言語でも 1 行です。モデルとして作成するならば、LED を 1 個点灯させればよいだけです。

大変複雑そうに見えるシステムも、細かく分析すれば、結局上記 4 行で説明できるものです。

この節で説明した燃料噴射制御は、皆さんが良く使うマイコンで容易にモデル開発ができるはず。チャレンジしてみたいはかがでしょうか。

8章 センサー

自動車は、車両基本性能の向上とエミッション低減（排出・放出）や安全性、快適性、利便性なども向上させるために、多くの技術的改良がおこなわれてきました。このような自動車の進化を裏方で支えてきたのが多くのセンサーです。環境・エネルギー問題に対応するために、センサーへの期待はますます大きくなっています。今や自動車1台当たり数十種類のセンサーがキーパーツとして使用されています。この章では、主な自動車用センサーの現状と今後の技術動向について概説します。

8章1節 自動車用センサーの技術的ニーズ

8. 1. 1. 環境の変化と制御システムの対応

今日の環境問題は、都市における大気汚染、水質汚濁、騒音といった局地的な問題にとどまらず、オゾン層破壊や酸性雨、地球温暖化現象など国レベルから地球的規模の広範囲な問題となっています。自動車の排出ガスが地球環境に悪影響を及ぼしていることは、周知の事実であり、地球環境保全に向けて自動車の排出ガスをクリーンにすることは早急に対処しなければいけない問題です。自動車の排出ガスは、給油時の燃料蒸発ガス及び排気管から放出されるガスの総和で表わされます。世界で最も厳しい自動車の排出ガス規制を実施しているアメリカ：カリフォルニア州における規制動向では、HC（ヒドロカーボン-炭化水素化合物）、CO（一酸化炭素）、NO_x（窒素酸化物）の排出量規制レベルは、最近の20年間で大変厳しいものになりました。低公害車の更なる普及が加速され、ZEV（ゼロエミッション車）に向けて、HEV（ハイブリッド電気自動車）の販売促進とともに、燃料電池を用いたEV（電気自動車）の研究が注目されています。エネルギー問題では、資源の枯渇問題から、省エネルギーすなわち自動車の燃費向上が今後も重要な課題です。日米欧各国にて表8のような自動車の燃費規制目標基準値を定め、更なる燃費向上への取り組みが進められてきました。

表 8 燃費向上目標 (出典：デンソー)

単位：km/L			
車種	ガソリン車	ディーゼル車	備考
日本	12.3 (1995 実績) 15.1 (2010 予測)	10.1 (1995 実績) 11.6 (2005 予測)	乗用車＋ 2.5t以下貨物車
米国	11.6 (2001)	11.6 (2001)	CAFE 27.5mile/gallon
欧州	14.1-14.5 (2003) (CO ₂ 165-170g/km) 17.1 (2008) (CO ₂ 140g/km)	15.7-16.1 (2003) 19.0 (2008)	各国の試験走行 モードは異なる

8. 1. 2. センサーの技術的ニーズ

次の図に示したさまざまな自動車の技術課題のなかで、特に地球環境問題、エネルギー問題は、自動車関連技術者が克服すべき最大の課題といえます。地球環境保全及び省エネルギー・燃費向上は、排出ガス浄化システムやガソリン直噴希薄燃焼システムのような制御システムで対応してきており、パワートレイン制御システムの体系に位置づけられます。一方、安全性、快適性、利便性向上のニーズは、車両制御、ボディー制御、情報通信システムと関連づけられます。車載センサーは、これらの制御システムに必要な検出感度・検出精度・応答速度などの機能を自動車の過酷な使用環境において維持することが要求されて、耐熱性や耐振動性などの信頼性確保も不可欠です。また大量生産される自動車の部品という観点から製造のしやすさや経済性も重要な要件として挙げられます。

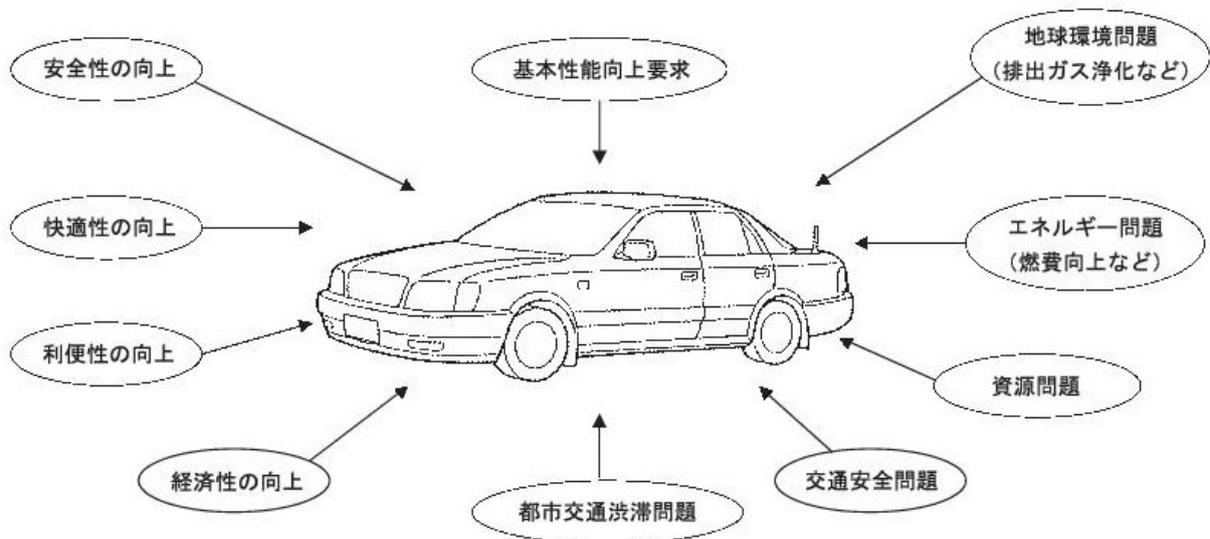


図 146 自動車開発技術の課題 (出典：デンソー)

8. 1. 3. 検出対象の拡大と複合化

より精密かつ高応答の制御システムを成立させるためには、センサーの技術レベル向上が必要不可欠です。センサー検出対象はより拡大しており、一つのセンサーが複数の情報を処理する複合センサーのニーズも高まっています。例えば、排出ガス浄化システムで用いられる排ガスセンサーは、排出ガス中酸素濃度を検出するものですが、排出ガス浄化システムのレベル向上のために、燃料リッチ領域（気化燃料ガスで燃料成分が多い領域）での未燃ガス（HC）濃度やNO_x濃度も同時に検出する複合センサーの開発が進められています。

8. 1. 4. センサー材料の変化

センサーに使われる材料は、金属やセラミックスの他に半導体の使用が多くなってきています。例えば、車体の速度変化を検出する加速度（G）センサーにおいても、ひずみゲージ材料として、従来の金属抵抗体からシリコン半導体へと変えることでゲージ率を50倍程度上げることが可能となり、センサー検出精度の大幅な向上につながっています。

8. 1. 5. 作動原理・変換機能の多様化

センサーを変換機能で分類すると、図のようになります。これは、検出したい外界の情報の種類で分けたもので、いずれも物理現象や化学現象を利用して電気信号に変換するものです。力学的センサーは、力学的諸量を電気信号に変換するものです。力学的諸量は、機械量と流体量に分類されます。機械量のうち、幾何学量の時間的変化を運動量と云い、質量・力・トルクなどは、力学量と呼ばれます。自動車用として、速度センサー、加速度センサーや圧力センサー、空気量センサーなどが数多く用いられていますが、用途に応じてさまざまな作動原理・変換機能に基づく各種センサーが開発されています。例えば、空気量センサーは吸入空気量を測定するものですが、従来の機械式センサーに対して、熱式エアフローメータ、半導体式吸気圧センサーなどが開発・実用化されています。

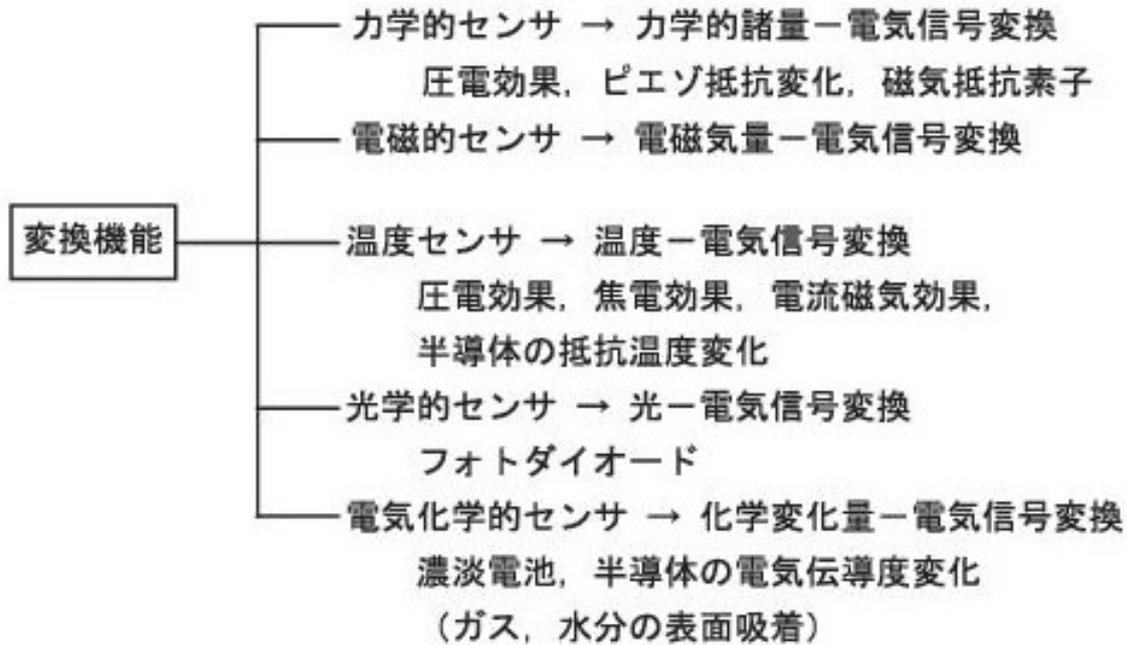


図 147 センサー分類 (出典：デンソー)

電磁的センサーは、電磁気量を電気信号に変換するもので、電磁ピックアップの他に、磁気抵抗効果、ホール効果などを利用した半導体磁気センサーなどが自動車用の車速センサーや回転角センサーに用いられています。温度センサーは、温度を電気信号に変換するものです。使用方法から接触式と非接触式に分類されますが、自動車用のものは、大半が直接物体に接触して検出する接触式を用いていて、サーミスタ（温度により抵抗変化する半導体）素子やバイメタルを利用するケースが多くみられます。非接触式では、乗員センサーとして焦電効果を利用する赤外線センサーが適用例として挙げられます。

光学的センサーは、光を電気信号に変換するもので、光と半導体との相互作用による電子-正孔対の発生を検出する半導体光センサーが用いられています。自動車用としてはエアコン制御用にフォトダイオードを適用した日射センサーが使われています。以上の力学的センサー、電磁的センサー、温度センサー、光学的センサーは、物理変化量を電気信号に変換するため総称して物理センサーと呼ぶのに対して、電気化学的センサーは、化学変化量を電気信号に変換するものです。自動車用としては、排出ガス中の酸素濃度を検出するためのジルコニアセラミックス製固体電解質素子を用いた O₂ センサーが代表例として挙げられます。

8. 1. 6. 信号処理のインテリジェント化

センサーデバイスで発生する電気信号は、微弱であったり、対象となる物理量や化学量

以外のパラメータの情報を含んでいたり、非直線的であることが多いため、センサーの信号処理が必要です。センサー信号処理の目的は、センサーから発生する電気信号を伝送またはコンピュータが処理できる電気レベルに増幅したり、不必要な情報を除去したり、非直線性の補正を行うことなどにより、情報の質を高めることにあります。センサーの本来の機能である物理量・化学量の変化を電気信号へ変換する処理以外に、その信号を増幅、補償する信号補正処理、信号を扱いやすい関数に変換する演算処理、制御のための情報処理、制御対象に信号を送るための制御処理が必要な処理機能として挙げられます。

センサーから伝送された信号をコンピュータが信号処理するだけでなく、最近ではセンサー側にマイコンを搭載してインテリジェント化させたスマートセンサーも多く開発されています。これにより、センサー自身がより高度な信号処理を受け持つことができ、センサーの自己診断や自動較正、データ記憶、複合情報の提供などが可能となっています。自動車用センサーでは、このようなセンサーの集積化・多機能化によるインテリジェント化の傾向が更に進むことが容易に予想されます。集積化は、多素子化、増幅器との一体化、信号処理機能の一体化、別機能素子との一体化などによって進められ、小形化、軽量化、新機能化、高性能化、高信頼性を図るものとなります。

下に示した静電容量式加速度センサーは、マイクロマシン技術を用いて、検出素子と信号処理 IC をスタック構造として集積化したスマートセンサーの一例です。この他にも半導体式吸気圧センサー（空気量センサー）や、IC 磁気センサーが例として挙げられます。

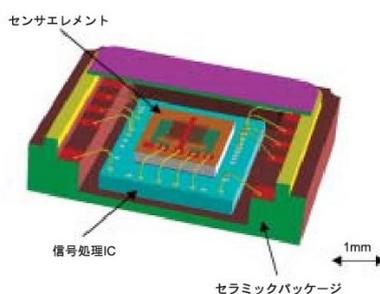


図 148 静電容量式加速度センサーの例（出典：デンソー）

8章2節 自動車用センサーの分類

表に自動車用センサーをパワートレイン制御、車両制御、ボディー制御、情報通信の4つの制御システムに用いられる主なセンサーについて、変換機能別の分類を示します。以下、これらの制御システムに用いられる主なセンサーについて説明します。

表 9 センサーの分類 (出典：デンソー)

対応制御システム			パワートレイン制御	車両制御	ボデー制御	情報通信	
自動車の技術開発課題			地球環境保全, 省エネルギー		基本性能向上		
ニーズ			排出ガス浄化, 燃費向上	安全性向上, 快適性向上		利便性向上	
システム例			<ul style="list-style-type: none"> ・ガソリン燃料噴射制御 ・空燃比フィードバック制御 ・希薄燃焼制御 ・ガソリン直噴制御 ・ガソリン点火時期制御 ・ディーゼル燃料噴射制御 ・アイドル回転数制御 ・オートマチックトランスミッション 	<ul style="list-style-type: none"> ・サスペンション制御 ・定速走行制御 (オートドライブ) ・アンチロックブレーキ (ABS) ・トラクションコントロール ・走行姿勢制御 (4WS, VSC) 	<ul style="list-style-type: none"> ・オートエアコン ・空気清浄 ・エアバッグ ・デジタルメータ ・ライト ・前方後方監視 	<ul style="list-style-type: none"> ・パーソナル無線 ・ナビゲーション ・自動車電話 ・自動車TV ・VICS対応ナビゲーション 	
物理 センサ	力学的 センサ	距離		・レーザーダ	・バックソナー ・コーナソナー	・超音波 ・レーザーダ ・CCD	
		位置・角度	・スロットル開度 ・アクセル開度	・ステアリング ・車高 ・スロットル開度	・エアミクスダンパ ポテンショ		
		加速度・振動	・ノック	・加速度	・加速度 ・セイフィング ・衝突検知		
		角速度		・角速度 (ヨーレート)		・ジャイロ	
		圧力	・エンジン吸気圧 ・大気圧 ・タンク内圧	・燃料圧 ・燃焼圧	・ブレーキ圧	・エアコン冷媒圧 ・タイヤ空気圧	
		流量	・空気量 (エアフローメータ) ・目詰り				
	電磁的 センサ	位置・ 回転速度	・車速 ・クラック角 (位置) ・カム角 (位置) ・回転数	・車輪速 ・車速	・オートマチック回転 ・車速 ・プロペラシャフト	・地磁気 ・車速	
		電波			・キーレス用アンテナ	・ラジオ用アンテナ ・GPS用アンテナ ・VICS用アンテナ ・自動車電話用アンテナ	
	光学的 センサ	光	・エンジン着火時期		・日射 ・光 (コンライト)		
	温度 センサ	温度	・エンジン水温 ・燃料温 ・吸気温 ・排ガス温		・内気温, 外気温 ・エバポレータ出口温 ・水温 ・オートマチック油温 ・乗員		
化学 センサ	電気 化学的 センサ	ガス濃度	・O ₂ (酸素) ・A/F (空燃比) ・HC, NO _x		・スモーク (車内煙) ・ガス (CO) ・湿度		

8. 2. 1. パワートレイン制御システムのセンサー

パワートレイン制御とは、燃料噴射制御、点火時期制御、アイドル回転数制御などの、各種エンジン制御ならびにトランスミッションなどの動力伝達系の制御を合わせたものであり、エンジンを最適状態で作動させるものです。

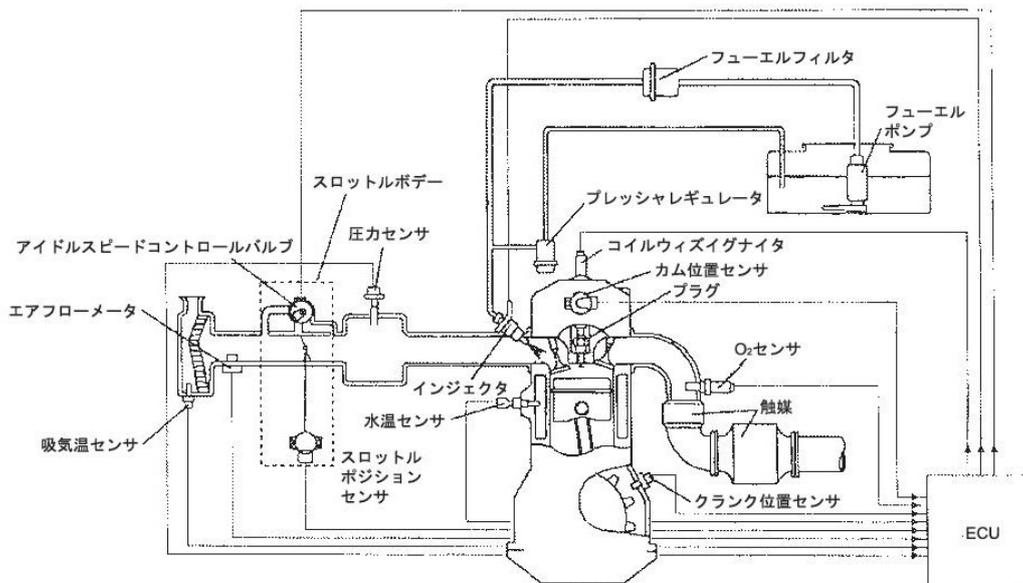


図 149 ガソリンエンジンの制御システム (出典：デンソー)

特に環境問題、エネルギー問題への対応は、例えば、上図に示すようなガソリンエンジン制御システム技術によって、排出ガス浄化や燃費向上が進められてきました。燃料噴射制御では、吸入空気量を基準として空燃比を制御することから、スロットル開度とエンジン回転数から空気量を推定します、空気量センサーは、この吸入空気量を正確に測定するものです。熱式エアフローメータの他にベーン（機械）式センサーや半導体式吸気圧センサーなどが用いられています。インジェクタの燃料噴射量は、空気量から決められますが、両者の計量誤差は、理論空燃比からのずれを生じさせてしまいます。このずれを検出して三元触媒のウィンドウの中心に入るように噴射量を補正するのが空燃比フィードバックシステムと呼ばれるもので、この補正信号をコンピュータへ送るのが、排出ガス中の酸素濃度を検出する O₂ センサーです。また、O₂ センサーは OBD（On Board Diagnosis）で触媒などのエミッション関連部品の劣化を検出して自己診断するための情報提供の役割も担っています。

より厳しくなる自動車排出ガス規制に対応するために、従来のコップ形状の検出素子の O₂ センサーから早期活性（速熱性）の点で構造上高いポテンシャルを有する積層形状の O₂ センサーが開発・実用化されました。更に理論値からの空燃比のずれ量を検出する A/F センサーと O₂ センサーを組み合わせる精密空燃比制御システムにおいても早期活性のために積層 A/F センサー化が進められています。ガソリンエンジンの点火時期制御でも、排出ガス浄化と燃費向上に寄与しています。エンジン回転数、ピストン位置検出を行う回転角センサーであるクランク角センサーやカム角センサーは、従来の電磁ピックアップから制約の少ないホール効果や強磁性体磁気抵抗素子（MRE）を利用する磁気センサーへと進化しています。この他にも、圧電効果を利用するノックセンサーなどが用いられています。

ディーゼルエンジンの電子制御は、スロットル開度センサー、回転数センサー、クランク角センサーなどを用いて、燃料噴射量、燃料噴射開始タイミング、吸気絞りなどを制御しています。コモンレールタイプ（蓄圧式）では、180MPa までの高圧を検出するコモンレール圧センサーが用いられ、ほぼ全運転領域で一定の噴射圧力が得られるため制御性が向上して、スモーク排出、騒音、振動などを改善しています。

電気自動車では、メインバッテリーの充放電電流を検出する電流センサーが特徴的です。また、エンジンの電子スロットル制御と同様に、ドライバーの直接のアクセル操作によらない駆動方式をとっており、アクセルセンサを介してモーターへ情報を送り制御しています。

8. 2. 2. 車両制御システムのセンサー

車両制御とは、自動車の基本性能である「走る」、「曲がる」、「止まる」という動作を電子制御して、応答性、操縦安定性、快適性などを改善するシステムを指します。車両制御には、サスペンション制御、ステアリング制御、ABS 制御、定速走行制御、トラクション制御、四輪操舵（4WS）制御、車両姿勢制御（VSC）システム、更には、それらを組み合わせて新しい機能を生み出すものなどがあり、車高（ハイト）センサー、ステアリングセンサー、車速センサー、車輪速センサー、加速度センサー、角速度（ヨーレート）センサーなどが用いられています。このうち、車輪速センサーの情報は、車両制御システムの中で、ABS、トラクション制御、4WS のシステムで共通に使われています。このように一つのセンサーの情報が複数のシステムで使われることも珍しくありません。

ABS（Antilock Braking System）制御は、主に車両の制御時に起こる車輪のロックを防止し、車両の操舵安定性を確保するものです。車輪ロック状態は、車体速度と車輪速度を比較して間接的に検出します。ただし、車体速度を車輪速度から推定する方法では、タイヤの滑りがあるために正確さが損なわれてしまいます。ABS の検出精度を上げるためには、タイヤの滑りの影響を排除できる車輪速度以外の信号による正確な対地車体速度センサーが必要です。定速走行制御は、クルーズコントロールとも呼ばれており、設定された車速を保つように、スロットルバルブ開度を制御するものです。車速センサーの一例としては、MRE を用いたミッションの回転数を検出するものがあります。

8. 2. 3. ボディー制御システムのセンサー

ボディー制御とは、自動車の安全性や快適性向上のための、エアバッグ、オートエアコン、デジタルメータ、ライト制御、後方障害物検出装置、ワイパー制御、パワーウィンド

ウ、ドアロックなどの電子制御を指します。エアバッグは、自動車の衝突を加速度センサーで検出して、ステアリングホイール内やダッシュボード内などに装着されたバッグに窒素ガスなどを急速充填することにより膨らませ、乗員を保護するシステムです。

オートエアコンは、温度設定スイッチにより希望の温度が設定されると、車室内をその設定温度に近づけるように各種温度センサーや日射センサーなどで制御するシステムで、吹出し口の温度制御、風量制御、吹出し口・吸込み口の制御、コンプレッサの制御などを行い、快適性を向上させています。また、湿度、匂いなども、エアコンの制御対象となりますが、狭い居住空間を乗員数に応じていかに均一な環境に保つかが課題となります。

聴覚的な快適性では、車自体が出す音や外部から入ってくる不要な音を打ち消すように、雑音に対して逆位相の音を重畳するノイズキャンセラーにて対応する方法があります。

自動車に乗っている乗員の居住空間の快適性向上は、人の感性をいかに測定するかがポイントです。感性に影響を及ぼす何の因子を、どのようなセンサーを用いて測り、得られた信号でどのような制御を行うかは、個人の嗜好差もあるために一般解があるわけではなく、かなり難しい面があります。しかし、例えば、温度に対する感性は、外気に曝された皮膚の温度を赤外線センサーで検出して温度制御を行うなどの試みもおこなわれています。

快適性の更なる向上は、人の感性をも測定可能なセンサーの開発がキーテクノロジーの一つです。

8. 2. 4. 情報通信システム系のセンサー

情報通信システムとは、車外との通信により得られたデータなどにより乗員にさまざまな情報を提供するもので、ナビゲーションシステムや自動車電話が代表的なものになります。

ナビゲーションシステムは、自動車の現在位置や目的地への経路を表示装置の地図上に表示するもので、人工衛星からの信号による位置測定が可能な GPS 方式が一般的です。電波を受信する GPS アンテナの他に、車両の方位を求める地磁気センサー、回転角速度を検出するジャイロ、車速センサーなどが用いられます。位置情報に加えて、道路の渋滞情報、交通規制情報、駐車場情報なども得られるようになり、システムの活用範囲が拡大されてきました。ナビゲーションシステムや定速走行制御システムは、交通輸送システムへの情報通信技術の利用によって支えられています。

この情報通信技術の高度利用の研究は、1980年代から日本、米国、欧州で活発化し、国内では1980年代後半からカーナビゲーションシステムの商品化が始まって、新たな道路交通情報提供サービスの可能性を示しました。情報通信システムは、現在国家プロジェクトとして推進中の ITS (Intelligent Transport Systems=高度道路交通システム) 体系の

中で更に進化しています。ITS では、道路と車（路車間）または車と車（車車間）を通信により有機的に結びつけて、エレクトロニクス、情報処理、通信、制御技術を交通輸送分野に総合的に適用し、安全・快適・効率的で、環境にやさしい道路交通システムの実現を目指しています。みなさんご存知の ETC サービスも ITS の一環です。ITS の推進には、道路交通に関する公共的交通機関、自動車産業、電機関連産業、情報通信産業、交通輸送産業が、さまざまな立場でかかわっています。このような広がりをもつ ITS を円滑、着実に発展させるためには、公的機関と民間企業の密接な協力が不可欠です。自動車技術の目覚ましい進歩に比べて、遅れていた道路技術の差を ITS が埋めるように、次世代道路交通網実現のために提案されたシステムなのです。

ITS 技術分野を大きく分けると、情報網の整備と運転操作の自動化の二つに大別されます。路側機（ビーコン）と車載機とで構成される路車間通信などの情報体系や社会インフラの整備と、自動車技術のレベル向上で対応するものです。ドライバーサポートシステムとして、ABS、トラクションコントロールなどの駆動・制動系制御や、レーザレーダセンサーによる定速走行制御（レーダクルーズコントロール）及び車間距離制御システム、トランスミッション制御などは、ドライバーのアクセル、ハンドル、ブレーキ操作による「走る」、「曲がる」、「止まる」の性能を、走行環境に適応して向上させ、安全かつ快適な運転の実現を間接的に支援しています。自動車の安全性の飛躍的高度化をうながすために、国土交通省が提案している先進安全自動車（ASV=Advanced Safety Vehicle）プロジェクトのイメージを示します。

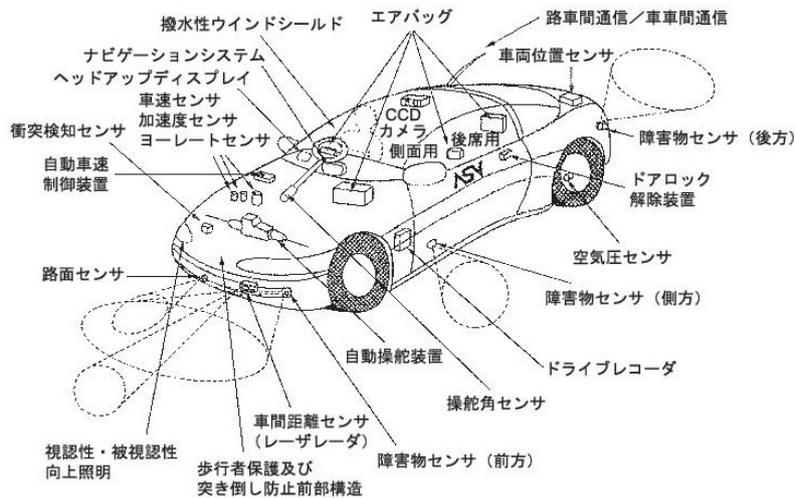


図 150 AVS のイメージ (出典：デンソー)

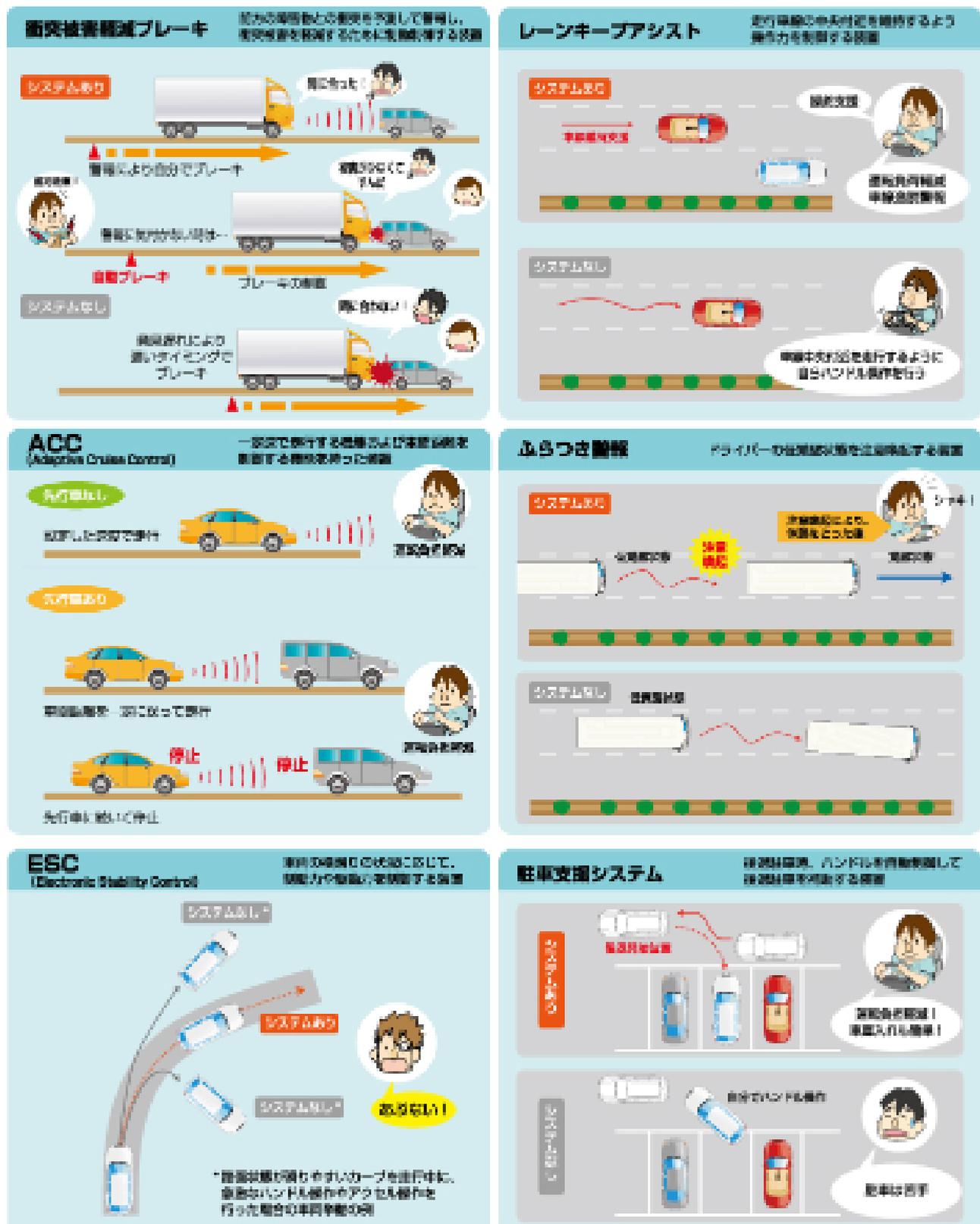


図 151 すでに実用化された AVS の例 (出典：国土交通省)

将来的にはドライバーの知覚・判断のプロセスも支援して安全かつ快適に総合制御する自動運転システムへと進化させていくことが期待されています。

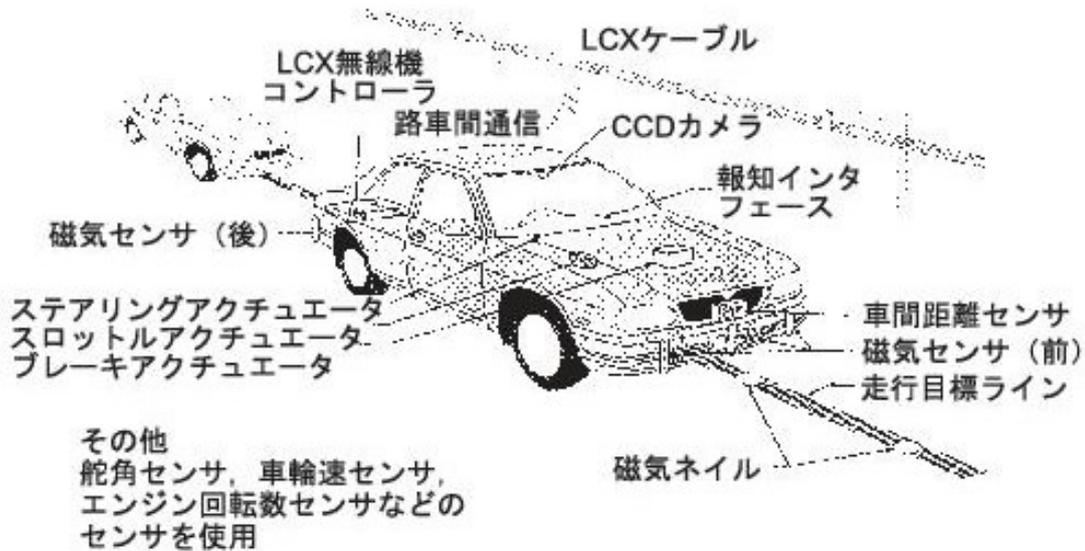


図 152 自動運転システムのイメージ (出典：デンソー)

上の図は自動運転システムのイメージです。運転操作のプロセスは、外界からの情報を収集する「知覚」、それに基づき高度な情報処理を行う「判断」、そしてその結果に伴い行う「操作」の一連の流れがあります。すでに解説したさまざまなドライバーサポートシステムは、この知覚、判断、操作のプロセスを支援して、ドライバーの負担を軽減し、ミスをカバーして、ヒューマンエラーによる交通事故などを未然に防止することを狙いとしています。ドライバーの目耳に相当するセンサー、頭脳に相当するコンピュータ、手足に相当するアクチュエータの調和が大切で、カーエレクトロニクスの更なる発展がこのシステムを成立させる必要条件であることは言うまでもありません。ITS 社会実現に向けて、自動車と道路側の双方で、情報収集・伝達のために、センサーが“エレクトロニクスの感覚器官”としてとりわけ重要な役割を担うこととなります。

8章3節 GPS センサーの簡単な実験

位置情報を得るために使われる GPS センサーは、現在入手するのも容易で、取り扱いの情報も WEB 上に多く公開されています。この節では、図に示すような GPS センサーを使って、位置情報を衛星から受信してその結果を高空からの写真に重ね合わせる実験をしてみます。



図 153 GPS センサー

8. 3. 1. GPS センサーの信号

このセンサーは 2cm 角ほどの大きさで、下図のような PIN アサインになっています。PIN アサインを見ると、PIN3~6 に TTL レベルと RS-232 レベルのシリアル通信信号が取り出されています。PIN2 を見ると電源は 3.3~6.0VDC となっていて、USB または乾電池での駆動もできそうです。このセンサーを使えば、RS-232 レベルで PC と、または TTL レベルでマイコンと直接接続ができることが分かります。

PINOUT DESCRIPTION

Pin Number	Signal Name	Description
1	Ground	Power and signal ground
2	Power	3.3V ~ 6.0V DC input
3	Serial Data In 2	Asynchronous serial input at RS-232 level, to input command message
4	Serial Data Out 2	Asynchronous serial output at RS-232 level, to output NMEA message
5	Serial Data In 1	Asynchronous serial input at TTL level, to input command message
6	Serial Data Out 2	Asynchronous serial output at TTL level, to output NMEA message

図 154 GPS センサーの PIN アサイン

8. 3. 2. NMEA0183 フォーマット

GPS センサーは、人工衛星からの電波を受信して測位情報を通知してくれます。GPS センサーはどのようにデータを出力するかというと、NMEA0183 フォーマットという様式が定められていて、この様式に合わせたデータをシリアル通信で出力します。この様式は、要約すると次のようなものです。

- [1] CSV 形式である
- [2] 先頭は \$ GP で、メッセージ ID が続く
- [3] \$ GPGGA → UTC (※) 測位時刻
- [4] \$ GPRMC → 緯度・経度・進行方位・速度
- [5] \$ GPGSA → 捕捉している衛星番号など
- [6] \$ GPVTG → 進行方位・速度

※世界標時が UTC : Coordinated Universal Time (協定世界時) です。
JST (日本標準時) は UTC + 9 時間となっています。

実際の受信データは次のようになります。

```
$GPGGA,080709.616,3553.1706,N,13955.6168,E,1,06,1.6,24.6,M,36.5,M,,0000*68
$GPGSA,A,3,06,16,03,32,21,14,,,,,,,,,2.9,1.6,2.5*3C
$GPGSV,3,1,11,19,43,069,,16,38,272,29,12,35,053,,06,28,208,38*75
$GPGSV,3,2,11,14,22,171,35,21,18,121,30,23,15,318,,03,10,216,37*7B
$GPGSV,3,3,11,32,06,257,27,02,03,042,,20,00,280,*44
$GPVTG,237.8,T,M,000.0,N,000.0,K,A*03
$GPRMC,080710.616,A,3553.1706,N,13955.6168,E,000.0,237.8,210413,,,A*68
```

これをテキストファイルとして保存すれば、移動経路や現在地を容易に地図上に描くことができます。

8. 3. 3. マイコンの選定

PC と接続するのであれば、以上の情報で容易に GPS センサーのデータを受信し、保存することができるでしょう。今回は、外に持ち出し、自動車や自転車、徒歩などで、移動してその経路を SD カードに記録しておき、後でそのファイルを読み込んで地図に重ねる方法を採用することにしました。

そのためにシリアル通信ができて SD カードが使える乾電池駆動のできるマイコンをつかいます。今回使用したのは、Arduino というマイコンです。この Arduino も WEB 上に多くの公開情報があるので、ひょっとすると同じようなことを行っている方が世界のどこかにいるかもしれません。開発環境は WEB からダウンロードしてインストールしておきます。WEB ページは「Arduino」で検索すると容易に見つかります。開発言語は C++だと思ってください。ほんの少し違うようですが、ほぼ同じです。

作成したプログラムは、IDE から USB ケーブルを使って Arduino に書き込みができます。

8. 3. 4. センサーとマイコンの接続

センサーと Arduino の接続は、とても簡単です。ジャンパー線 3 本で下図のように配線します。

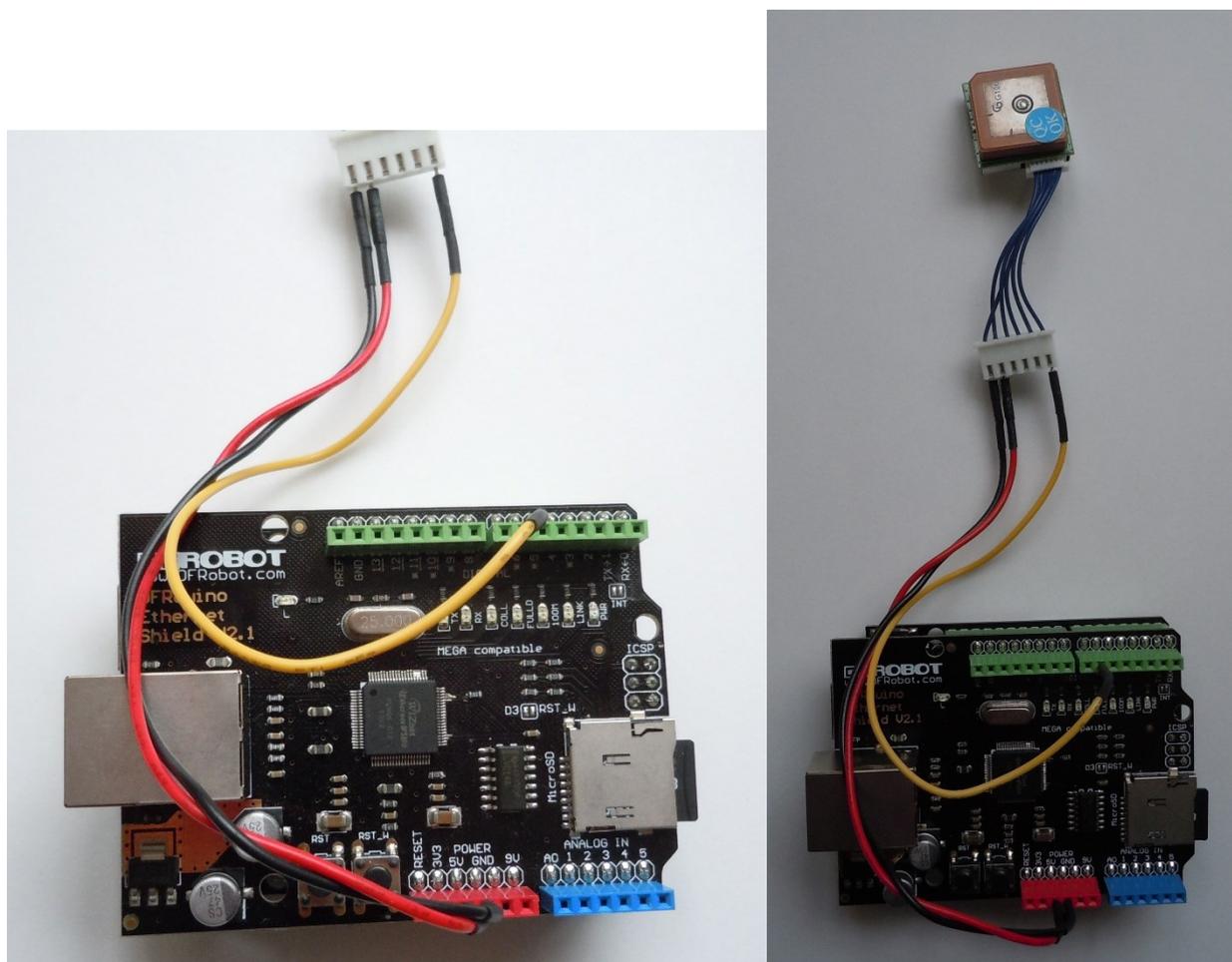


図 155 GPS センサーと Arduino の接続

GPS センサーのコネクターの PIN は、図の左側（黒のジャンパー線）が PIN1 (GND) です。黄色のジャンパー線は PIN6 (TTL レベルのシリアル OUT) です。これを Arduino

の 6 番ソケット(ソフトウェアシリアル Rx)に接続します。

GPS センサーの電源は Arduino 経由で赤いジャンパー線で供給します。Arduino 自身の電源は USB ケーブルから乾電池で供給することとしました。

上の図、左側の青色の PIN ソケットのすぐ上に SD カードドライブがあり、MicroSD カードをセットしてあります。Arduino 本体には SD カードドライブがありませんので、SD カードドライブを持つ Ethernet カードを Arduino 本体に重ねて使用しています。

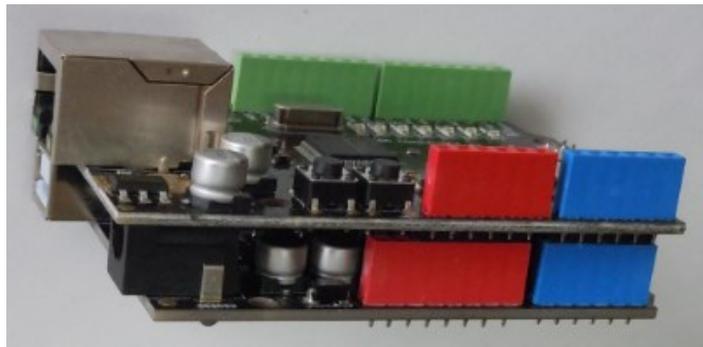


図 156 Arduino 本体の上に重ねた Ethernet カード

8. 3. 5. プログラム

SD カードへの書き込みは時間がかかりますので、受信したすべてのデータを SD カードに記録するには時間が足りません。(PC にシリアル接続していれば、全てファイルに保存できます。) 作成したプログラムは、次のようになりました。

```
#include <string.h>
#include <SoftwareSerial.h>
#include <SD.h>

#define CS 4 //Required Chip select
           //CS は、使用する SD カードドライブの仕様を事前に調査してください。
#define LED 13

SoftwareSerial gps( 6, 7 );

int i;
char msg[256];
```

```

void setup()
{
  pinMode(CS, OUTPUT);    //SD カード用チップセレクト PIN
  pinMode(LED, OUTPUT);  //LED PIN を出力
  gps.begin(9600);        //GPS 接続ポートを 9600bps で初期化
  Serial.begin(9600);     //シリアルポートを 9600bps で初期化
  Serial.println("--- GPS Logger ---"); //タイトルをシリアル出力

  if(SD.begin(CS)==true) {
    Serial.println("SD Card Activate."); //SD カードは有効です
    SD.remove("GPS_LOG.LOG"); //ログファイルを削除
  } else {
    Serial.println("SD Card Not Active!!!"); //SD カードが無効の場合メッセージを出力。
    digitalWrite(LED, HIGH);
  }
  delay(3000);
  digitalWrite(LED, HIGH); //LED をブリンク。
  delay(200);
  digitalWrite(LED, LOW);
  delay(200);
  digitalWrite(LED, HIGH);
  delay(200);
  digitalWrite(LED, LOW);
  delay(200);
  digitalWrite(LED, HIGH);
  delay(200);
  digitalWrite(LED, LOW);

  Serial.println("---> Logging Start!!...");
  i=0;
  Serial.flush(); //シリアルバッファを空にする
}

void loop()
{

```

```

do {
  if (gps.available()) {          //GPS からデータ有るか
    msg[i++]=gps.read();        //バイト受信
  }

  if (i > 254) break;           //バッファフルに近づいたら、自動的にループ離脱
} while(msg[i-1] != 0x0a);      //ラインフィードまで繰り返す
msg[i]=0x00;                   //受信データ最後を NULL にして文字列にする
if ((strncmp("$GPGGA", msg, 6)==0) || (strncmp("$GPRMC", msg, 6)==0)) { //センテンス選択
  Serial.print(msg);           //受信データシリアルで表示
}

digitalWrite(LED, HIGH);       //SD カード書き込み中は LED 点灯
File sdFile=SD.open("gps_log.log",FILE_WRITE); //SD カードにデータ保存
sdFile.print(msg);
sdFile.close();
digitalWrite(LED, LOW);
i=0;
}

```

実際にこれを使い、移動しながら計測すると、SD カードに移動の記録が保存されます。このファイルを後で PC に読み込んで、解析します。

8. 3. 6. 完成した GPS ロガー

完成した GPS ロガーは、食品用パックに入れました。GPS センサーは蓋の内側にテープで固定してあります。電池ケースは単三電池×4 本のもので USB コネクタがついています。USB ケーブルで電池ケースと Arduino を接続して SW を ON すれば、記録が始まります。

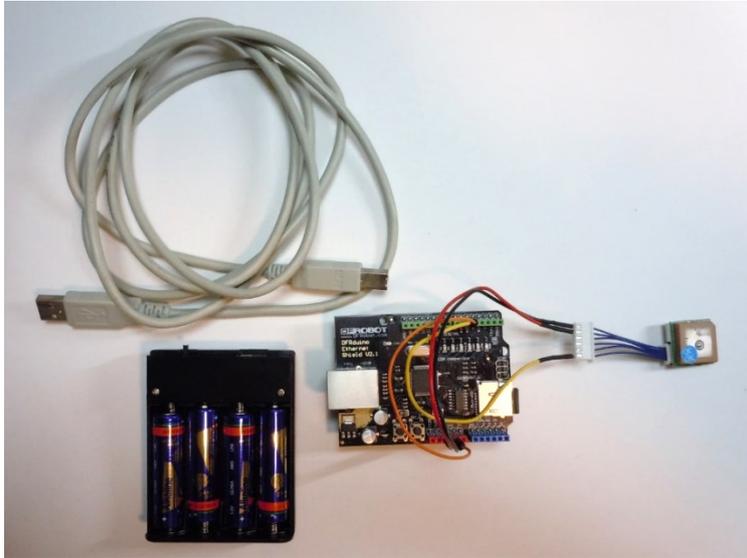


図 157 完成したユニット

※使用した GPS センサーは、衛星を捕捉するとインジケータ LED が点滅を始めます。経路を記録する際は、空が広く開いている場所で電源を入れ、衛星を捕捉し始めるまでしばらく待ち、それから移動を始めるようにします。



図 158 外ではケースに入れて使用

8. 3. 7. GPS ログを地図上に表示

SD カード内に記録された gps_log.log というファイルを使い、地図上に経路を表示します。表示には GoogleEarth を使います。Google の HP からダウンロードしてインストールしてください。準備ができれば、GoogleEarth を起動します。

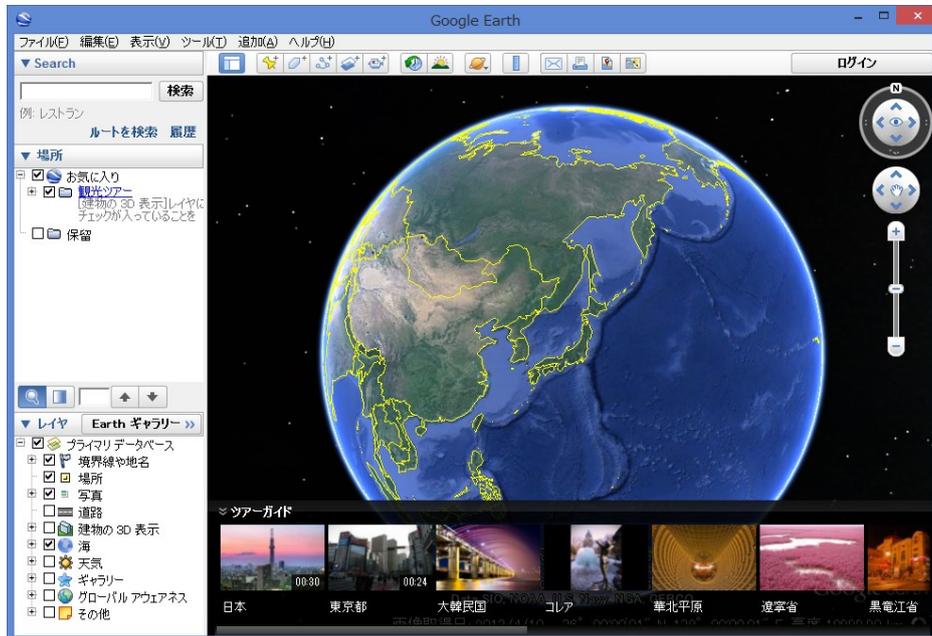


図 159 Google Earth (出典 : Google)

ツール→GPS を実行すると GPS インポートウィンドウが開きます。



図 160 GPS インポートウィンドウ

「ファイルからインポート」を選択して、インポートボタンをクリックすると、ファイル選択のウィンドウが開きますので、そこで先に取得しておいた GPS モジュールのデータファイル(gps_log.log)を指定します。

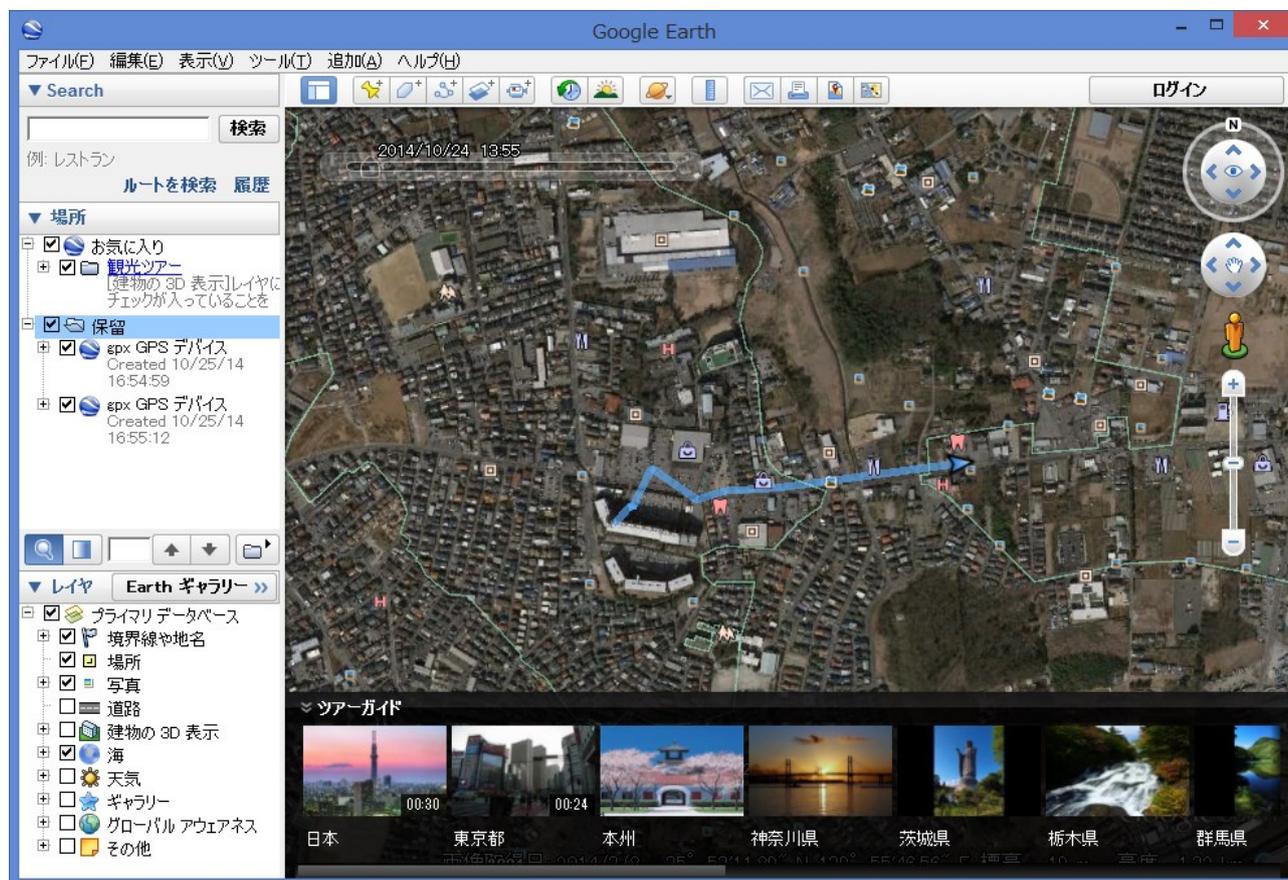


図 161 GPS データをインポートした

いかがでしょうか、上の図のようにうまく経路が表示されたでしょうか。

8章4節 この章のまとめ

自動車に関連するセンサーは、自動車の制御とともにあることが理解できたでしょうか。制御のニーズがありそのニーズにこたえるためにセンサーの開発が進められています。一般にセンサー自体は、値を電圧で出力するものがほとんどです。その場合は、マイコンを使い電圧値を計測し、使いやすい形に変換（演算）してデジタル値にします。また、直接デジタル値を出力するセンサーは、計測値をシリアル通信で取り出すものがほとんどです。その時に使われるシリアル通信は、通信の章で説明したものではなく、他のプロトコルによるものが多いのですが、プロトコルの違いをドライバーやライブラリソフトが吸収するようになっていて、意識する必要はほとんどありません。

実験では、簡単なシリアル通信をあまり意識せずに利用できるマイコンを使いましたが、ECUに直接かかわるセンサーユニットの場合、計測はもとより、通信部分をどうするかも大きな課題になります。

本書では、次の章でもセンサーを利用したシステム開発を行っていますので、参照してください。

9章 モニタリング表示

私たちの近くには、いたるところに LED や液晶表示が付いた機器があります。このような表示器は、機器の状態や環境を表示するモニタリング装置となっています。8章で解説した GPS センサーも自局の位置データを出力しているので、受信局の環境をモニタリングしていることになります。

この章では、モニタリングシステムを大づかみに理解して、マイコン、液晶表示器、センサーを使った装置を作ってみましょう。

9章1節 モニタリングシステム概要

下の図を見てください。単純なモニタリング装置です。センサー周囲の状況をマイコンが取得し、変換・加工などを行い LCD に文字で表示、または LED などによるインジケータ表示を行っています。

ここでマイコンが行わなくてはならないことは、①センサーを制御すること ②取り込んだ情報を表示しやすいように変換・加工すること ③表示すること の3つです。

一般的にマイコンは複数の入力ポートを持っているので、センサー1 つではポートが余ってしまい、もったいように思います。ですから同じ種類のセンサーを複数使ったり、他の種類のセンサーを使ったりして、表示器は共通に使う。などと考えます。すると、マイコンの処理が忙しくなって表示がちらついたり、種類の違うセンサーでは、データを取得するタイミングが違ったりして、プログラムが複雑になったりします。それは、開発に時間が掛かるということの意味します。



図 162 単純なモニタリング装置

では、次の図を見てください。センサーのつながっているマイコンは、データ取得専用です。この際、センサーから取り込んだデータの取扱いは置いておき、表示用マイコンに

表示の元になるデータを送っています。このマイコン間のやり取りは通信機能になります。通信機能のプロトコルとデータフォーマットを決めておけば、容易にセンサーの数を増やすことができます。図では 2 つのセンサーマイコンがつながっていますが、この通信部分を BUS 化したものが、通信の章で説明した CAN 等々です。

ということは、自動車であれば、この図で説明しているマイコンが ECU ということになります。

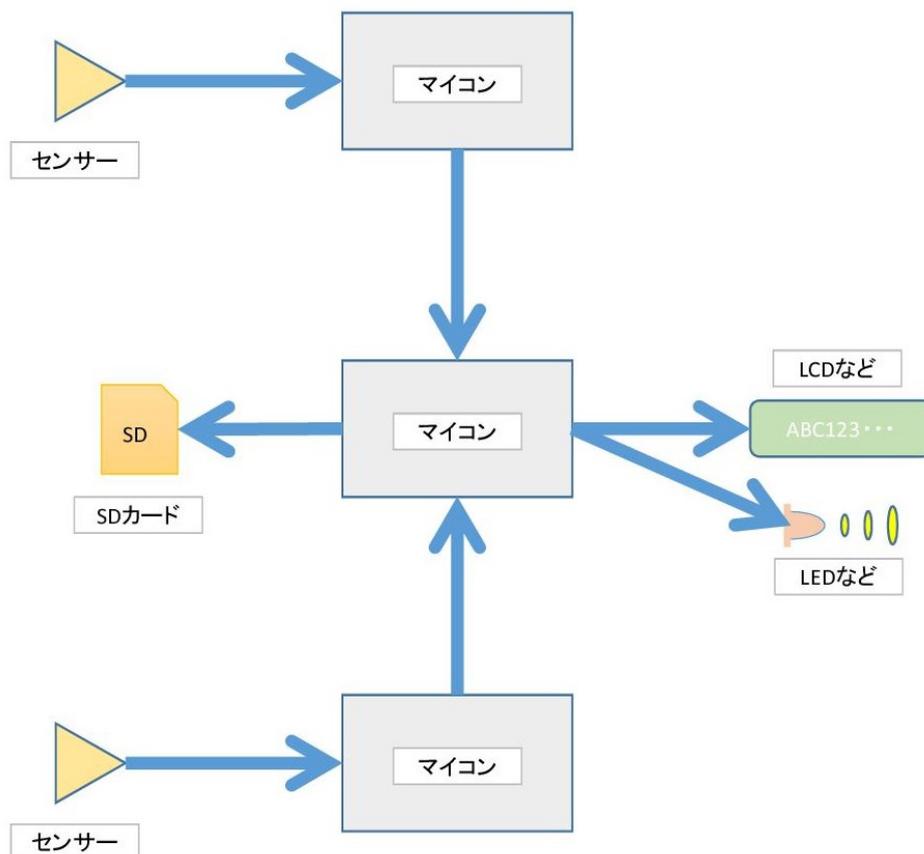


図 163 複雑なモニタリング装置

9章2節 求められる ECU の処理

センサー ECU では、取り扱うセンサーの種類によって、取込の手順が異なったり、取り込めるデータの範囲がいろいろであったりします。ですから、使用するセンサー毎にプログラムを開発することになります。たとえば、温度センサーも、計測の範囲が 0°C ~ の場合と氷点下 ~ の場合があります。温度も水温の場合や気温の場合、蒸気の場合や燃焼温度などを考えると、測定範囲がかなり違います。

センサーから取り込めるデータは、一般的に電圧で取り込むことが多いのですが、センサーの特性が直線の場合とそうでない場合があります。次の図を見てください。左側のセンサー出力は、入力（温度や湿度等）に比例して出力（電圧）が変化しています。これであれば、マイコンは A/D 変換で電圧を測定し、比例計算で対応する温度を求めることが容易ですが、右側のように非線形な特性を持つものでは、直線的に変化する領域に限って簡単に利用できることになります。測定範囲が利用目的に叶えば、これでも OK です。

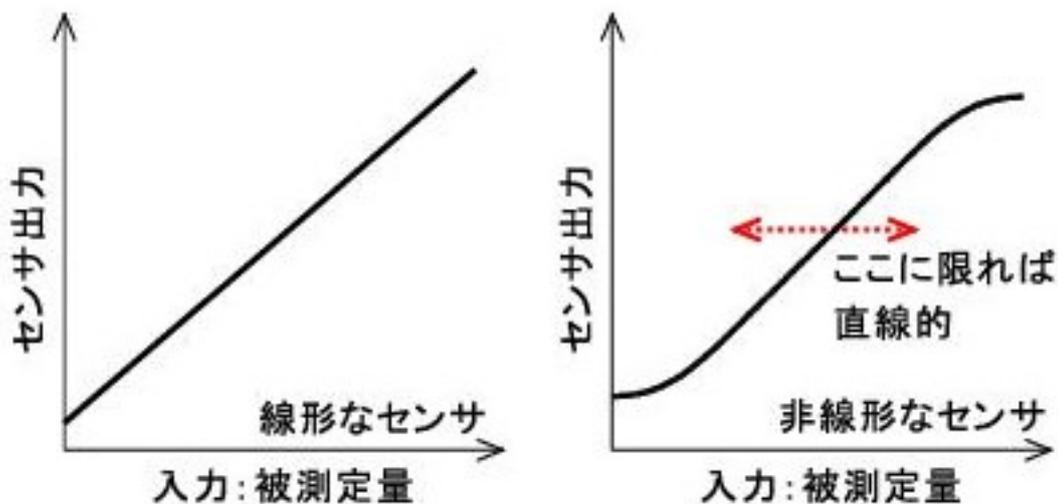


図 164 センサー特性

次の図は、湿度センサーの特性例です。湿度に対して出力が曲線的であり、さらに温度によっても出力がシフトしています。さらに、出力（縦軸）がインピーダンスなので、交流成分に対する抵抗値ですから、マイコン側は、測定の際、交流を作る必要があります。

センサーシステムのプランとして、この部分をソフトウェアで処理すれば、ソフトウェア開発費が大きくなり、ハードウェアで処理すれば、ソフトウェア開発は楽になりますが、ハードウェアの設計費・製作費・材料費などがかかります。（どちらが良いのでしょうか？エンジニアはここで最適な答えを出さなくてははいけません。）

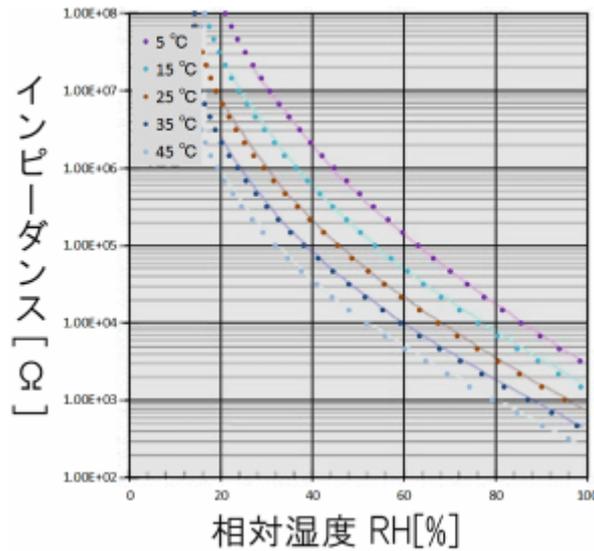


図 165 湿度センサーの特性例

表示器用 ECU は、使われる表示器が LED ならば ON/OFF の 2 値で出力することになり、判断の閾値をどうするかが問題になります。閾値をセンサー ECU で持つのか、表示器 ECU で持つのかによって、処理の負担が変わります。もし表示器がキャラクタ LCD の場合には、表示する文字列をセンサー ECU で作るか、表示用 ECU で作るかを決めなくては いけません。自動車内部の通信のトラフィック量は大変多くなっているから、できるだけ通信データ量を減らす方向になるでしょうから、文字列は表示用 ECU が作る こと になるでしょう。最近では、携帯電話に使われているようなグラフィック液晶を使った機器も 増えています。自動車も機械式メーターではなく、グラフィックス液晶を用いたものが多い になっています。

これから、そのグラフィック液晶表示器と各種センサーを使った、デジタルメータを 作成します。

9章3節 デジタルメータの開発

9. 3. 1. 表示器の選択

キャラクタ表示を行う LCD は、種類も豊富で安価で購入できるのですが、アナログメーター表示のようなことはできないので、今回はキャラクタもアナログメーターも表示できて、自由にデザインができるグラフィック液晶表示器を使います。

チャレンジングなイメージとしては、次の図のようなものです。



図 166 アナログメーターのイメージ

マイコンのアナログ入力に可変抵抗器（VR）を接続して、VR を回した時の電圧を測定して、その値に対応する位置に針を描くメーターをグラフィックスで描画します。上の図は燃料計ですが、意識としては、スピードメーターかタコメーターです。

今回は、携帯電話などに使用されている 1.8 インチ TFT カラー液晶表示器を使うことにします。（下図）

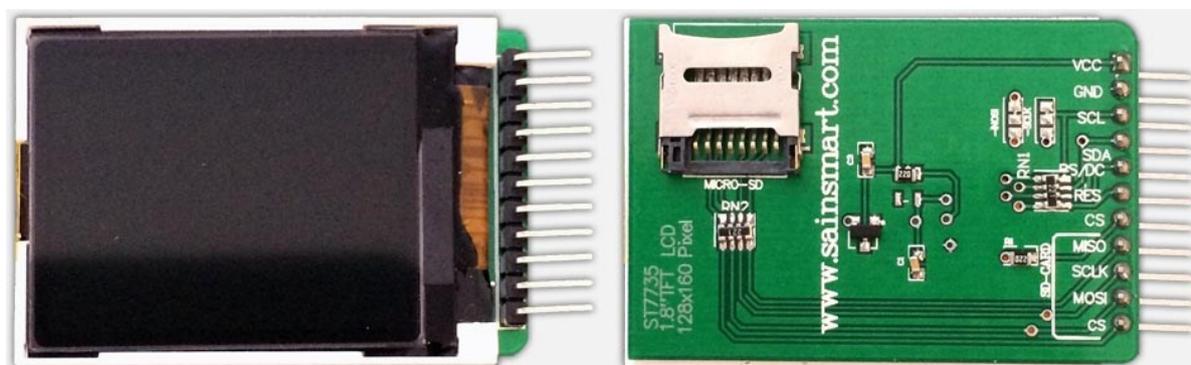


図 167 使用する液晶表示器

この表示器は、128×160 Pixel で裏側にマイクロ SD ドライブが付属していますが、今回は使いません。基板にはピンが固定されているので、そのままブレッドボードに挿して

使用できます。マイコンとの接続は、下図に示す PIN アサインで、SD カード以外の信号を利用します。電源は 3.3V でも駆動できるようですが、今回は 5V に接続して使用しました。

Group	Pin	Purpose
Backlight	VCC	+3.3 ... 5 Volt
	GND	Ground
Display	SCL	Display Clock
	SDA	LCD Data (SPI)
	RS/DC	Mode: Command/Data
	RES	Controller Reset
	CS (TFT)	Chip Select for display
SD-Card	MISO	Master In Slave Out (SPI)
	SCLK	Clock (SPI)
	MOSI	Master Out Slave In (SPI)
	CS (SD)	Chip Select for SD-Card

図 168 18”TFT カラー液晶表示器の PIN アサイン

9. 3. 2. マイコンの選択

マイコンは、GPS ロガーで利用した ArduinoUNO を使いました。Arduino のシリーズは、いろいろな種類が開発されていますが、どのモデルでもほぼ利用できます。純正品以外にサードパーティ製品も多く開発・市販されています。開発環境は、Arduino の HP から使用する PC の OS に合ったものをダウンロードします。ダウンロードしたファイルを実行すれば、特に指定することもなくインストールできます。

開発の際に、USB ケーブルを使用しますので、コネクタの形状を確認の上準備してください。



図 169 Arduino UNO と USB ケーブル (A オス - B オス)

9. 3. 3. VR (可変抵抗器)

スピードメーターイメージなので、アクセル代わりの可変抵抗器を使います。

今回は、 $5K\Omega$ のものを使用しました。5V を分圧してマイコンのアナログ入力に接続するので、抵抗値はどのようなものでも構いません。手持ちのもので、既に接続用のリード線が半田付けされたものを使用しましたが、半田付けを避けるなら、下の図右側の様な半固定抵抗器でも構いません。これであれば、ブレッドボードに直接挿して利用することができます。(ただし、固定用のリード線が長いものを選んでください、短いとブレッドボードに固定できません。)



図 170 VR (可変抵抗器と半固定抵抗器)

9. 3. 4. 照度センサー

屋内レベルでの明るさを測る目的で照度センサーを使います。測るといっても、精度は求めません。ここでは、図のようなフォトトランジスタを使用しました。



図 171 照度センサー (NJL7502L)

このセンサーは、データシートから 100Lux で $33\mu\text{A}$ の電流値となっています。直列に抵抗器を接続して、抵抗器の両端電圧を測定し、電流に換算してそこから明るさを求めます。(下図) 測定する照度と A/D 変換の分解能に応じて抵抗値は適宜変更します。今回は $1\text{k}\Omega$ としました。

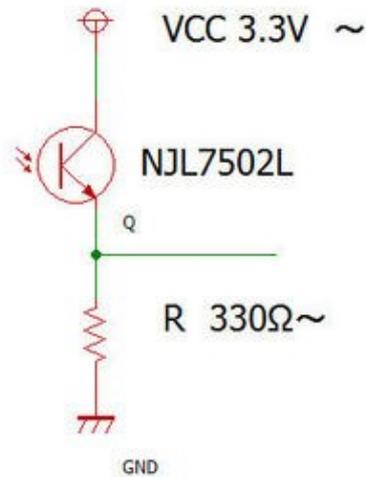


図 172 センサー回路 (GND~Q 間の電圧測定)

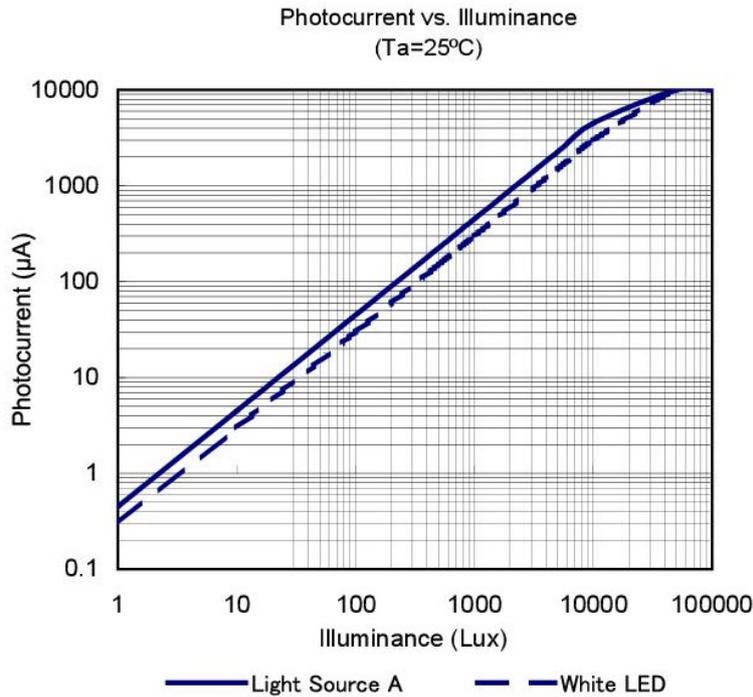


図 173 照度センサーの出力特性

ここで電圧＝抵抗値×電流値 です。センサーの出力特性は、下図のようにほとんどの領域でリニアですから比例計算ができます。使用するマイコンの A/D 変換の分解能は、5V を 1024 段階で計測するので、 $5V \div 1024 = 4.88mV$ です。A/D 変換の値を読んだとき 1 と読めれば、4.88mV ということです。センサー回路図のように抵抗値を 330Ω とすると、100Lux のときの電流値は $33\mu A$ ですから、 $33\mu A \times 330\Omega = 10890\mu V = 10.89mV$ です。100Lux は相当に暗いときの照度です。屋内の机上では無照明で 300～400Lux 程度。LED 卓上スタンドを点灯すると 1300Lux 程度になります。

比例計算で 6000Lux は、100Lux の 60 倍ですから、電圧値も 60 倍すると、 $10.89mV \times 60 = 653.4mV$ となります。これを分解能 4.88mV の A/D 変換器で測定すると約 134 となります。これでは A/D 変換のフルスケールの 1/10 程度しか使わないこととなってしまう、いくら「精度は求めない」と言っても、あまりにもったいないので、抵抗を $1k\Omega$ とします。

$$100\text{Lux} \rightarrow 33\mu A : 6000\text{Lux} \rightarrow 3000\mu A$$

の関係から、照度／電流 の比例係数を求めると

$$(6000 - 100) / (3000 - 33) = 1.99 \text{ [Lux}/\mu A]$$

分解能は、4.88mV ですから、照度は、

$$\text{照度} = \text{A/D 値} \times 4.88 \times 1.99 + 100\text{Lux}$$

となります。これで、A/D 変換器のフルスケールで 10000Lux 付近まで測定できます。照度センサーの出力電圧は、マイコンのアナログポート A1 に入力することにします。

9. 3. 5. 温度センサー

温度センサーは、LM61CIZ を使います。このセンサーは、測定範囲が広く全領域でリニアな特性で計算しやすい温度係数値となっています。

◇測定範囲： $-30^{\circ}\text{C} \sim 100^{\circ}\text{C}$

($-30^{\circ}\text{C} = 300\text{mV} \sim 100^{\circ}\text{C} = 1600\text{mV}$ 、 $0^{\circ}\text{C} = 600\text{mV}$)

◇温度係数： $+10\text{mV}/^{\circ}\text{C}$

◇動作電圧範囲： $+2.7 \sim +10\text{V}$

A/D 変換値から温度にする計算は、($-30^{\circ}\text{C} = 300\text{mV} \sim 100^{\circ}\text{C} = 1600\text{mV}$ 、 $0^{\circ}\text{C} = 600\text{mV}$)の特性から比例計算で求められます。このセンサーの出力は、マイコンのアナログポート A2 に入力することにします。

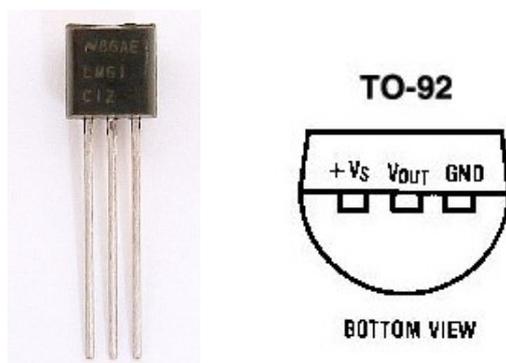


図 174 温度センサー (LM61CIZ)

9. 3. 6. 配線

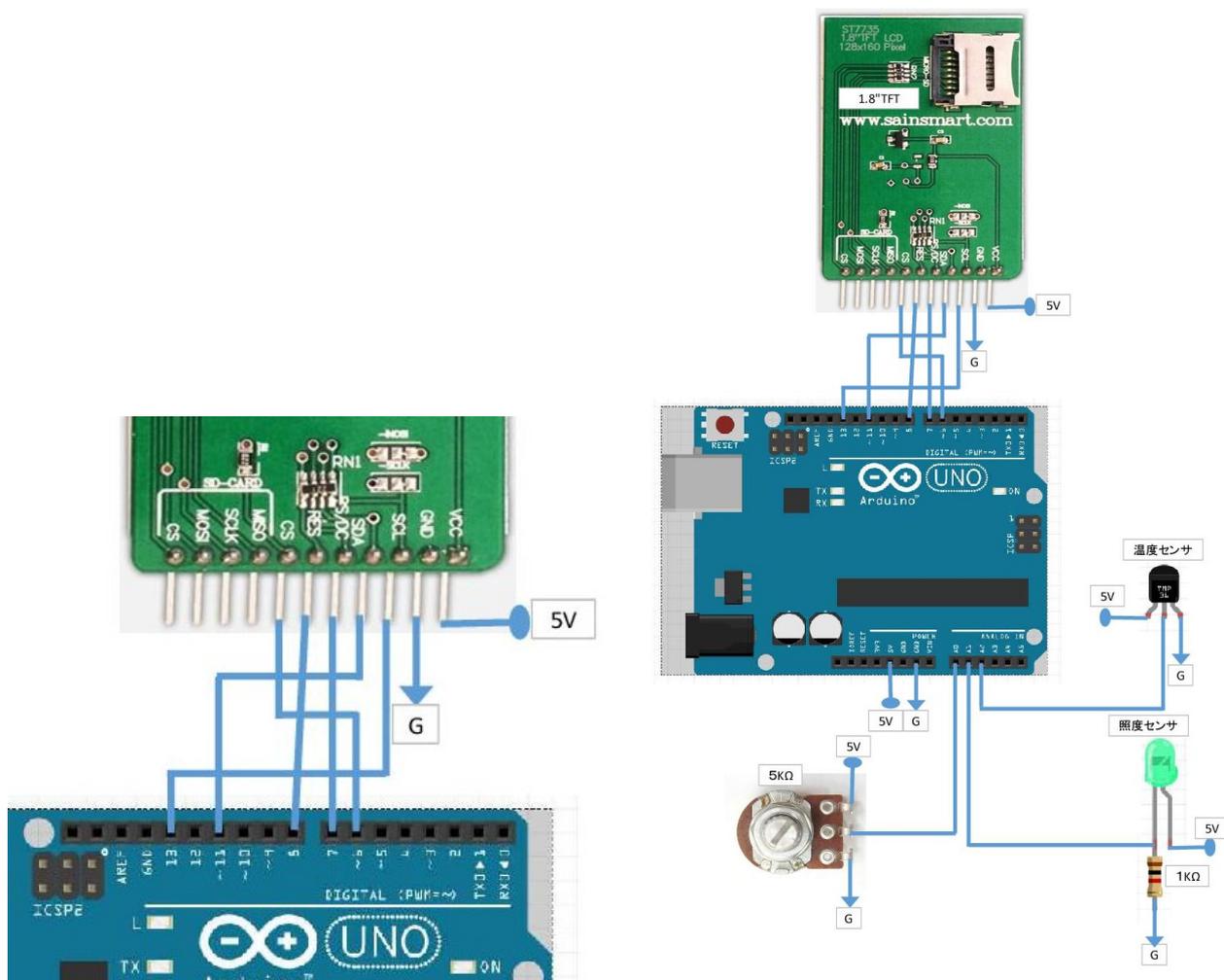


図 175 配線図

マイコンと TFT 液晶、センサー類との配線は上図のようにします。小さなブレッドボードを使用して、ジャンパー線で接続することで回路が出来上がります。

9. 3. 7. プログラム

プログラムの例を示します。

※-----※
 ※-----※

```
#include <ST7735.h>
```

```
#include <SPI.h>
```

```
#define sclk 13
```

```

#define mosi 11
#define cs 6
#define dc 7
#define rst 8 // you can also connect this to the Arduino reset
#define TEMP 2
#define LX 1
#define VR 0

// Color definitions
#define BLACK 0x0000
#define DARKGREY 0x38E7
#define GREY 0xC318
#define RED 0x001F
#define BLUE 0xF800
#define GREEN 0x07E0
#define DARKGREEN 0x0300
#define CYAN 0x07FF
#define MAGENTA 0xF81F
#define YELLOW 0xFFE0
#define WHITE 0xFFFF

#define R 50

#define LUX (9.711) //Lux Meter 分解能

int v=0; //as VR/4
int ov;
int r;
int x,y,ox,oy;
int lux_v, lux_ov;
int lux_i=0;
int ans, tv, temp, otemp; //for temperature

char s[4], os[4]; //for speed VR
char l[6], ol[6]; //for Lux
char t[3], ot[3]; //for temp.

```

```

// Option 1: use any pins but a little slower
ST7735 tft = ST7735(cs, dc, mosi, sclk, rst);

// Option 2: must use the hardware SPI pins
// (for UNO that's sclk = 13 and sid = 11) and pin 10 must be
// an output. This is much faster - also required if you want
// to use the microSD card (see the image drawing example)
//ST7735 tft = ST7735(cs, dc, rst);

void setup(void) {
  Serial.begin(115200);    //シリアルポートをボーレート 115200bps でオープンします。
  Serial.print("hello!"); //シリアル通信で"hello!"の文字列を送信します。
  tft.initR();           // initialize a ST7735R chip

  Serial.println("init"); //シリアル通信で"init"の文字列を送信します。

  tft.writecommand(ST7735_DISPON);

  uint16_t time = millis();
  tft.fillScreen(BLACK);
  time = millis() - time;

  r=tft.width/2-18; //meter radii. = needle length
  delay(500);

  //
  tft.fillScreen(BLACK);
  testdrawtext("Hello everybody. This is the Arduino Meter. You may find
that Arduino has powers for displaying analog meter with 1.8inch TFT GLCD.
", WHITE);
  delay(1000);

  //a single pixel center of screen

```

```

tft.drawPixel(tft.width/2, tft.height/2, GREEN);
delay(500);

tft.fillScreen(BLACK);
drawScale(BLUE, BLACK);
ov = 100;    //for needle
tft.drawString(82, 80, "km/h", GREY);

delay(1000);

lux_ov = 1000;
tft.drawString(84, 130, "Lux", GREY);
tft.drawString(84, 118, "Deg", GREY);
drawCopyright();

delay(1000);
}

void loop() {
  int val;
  long lux;

  val = analogRead(VR);

  v = val/4;

  if(v != ov){
    Serial.write(v);    //シリアル通信で、スピードの値を送信します。
    drawSpeed(v);
    drawNeedle(v,ov, BLACK, YELLOW);
    ov=v;
  }
}

```

```

if (lux_i++ == 10) {
    lux_i=0;
    lux_v = analogRead(LX);
    if (lux_v != lux_ov) {
        lux = lux_v * LUX + 100;

        drawLux(lux);

        lux_ov=lux_v;
    }
}

ans = analogRead(2);
tv = map(ans, 0, 1023, 0, 5000);
temp = map(tv, 300, 1600, -30, 100);
if(otemp != temp) {
    drawTemp(temp);
    otemp=temp;
}

delay(100);
}

void drawCopyright() {
    tft.drawString(40,145, "by ken.wiseman", DARKGREEN);
}

void drawTemp(int temp) {
    int n10;
    int n1;

    n10=temp/10;
    temp=temp-n10*10;
    n1=temp;
}

```

```

t[0]='0'+n10;
t[1]='0'+n1;
t[2]='¥0';

tft.drawString(68, 118, ot, BLACK);

ot[0]=t[0];
ot[1]=t[1];
ot[2]=t[2];

tft.drawString(68, 118, t, WHITE);

}

```

```

void drawLux(long lux) {
    int n10000;
    int n1000;
    int n100;
    int n10;
    int n1;

    // sprintf(s,"%d",v);
    n1000=lux/10000;
    lux=lux-n10000 * 10000;
    n1000=lux/1000;
    lux=lux-n1000 * 1000;
    n100=lux/100;
    lux=lux-n100 * 100;
    n10=lux/10;
    lux=lux-n10*10;
    n1=lux;

    l[0]='0'+n10000;
    l[1]='0'+n1000;
    l[2]='0'+n100;

```

```

l[3]='0'+n10;
l[4]='0'+n1;
l[5]='¥0';

tft.drawString(50, 130, ol, BLACK);

ol[0]=l[0];
ol[1]=l[1];
ol[2]=l[2];
ol[3]=l[3];
ol[4]=l[4];
ol[5]=l[5];

tft.drawString(50, 130, l, WHITE);
}

```

```

void drawSpeed(int v) {
    int n100;
    int n10;
    int n1;

    // sprintf(s,"%d",v);
    n100=v/100;
    v=v-n100 * 100;
    n10=v/10;
    v=v-n10*10;
    n1=v;

    s[0]='0'+n100;
    s[1]='0'+n10;
    s[2]='0'+n1;
    s[3]='¥0';

    tft.drawString(60, 80, os, BLACK);
}

```

```

os[0]=s[0];
os[1]=s[1];
os[2]=s[2];
os[3]=s[3];

tft.drawString(60, 80, s, WHITE);
}

void drawNeedle(int sp, int osp, uint16_t color1, uint16_t color2) {
    float rad;
    double xx,yy;

    tft.drawLine(tft.width/2, tft.width/2, ox, oy, color1); //まず、以前の針を消す

    rad=PI * sp / 360.0; //scaling fucter = 0.5
    x = tft.width/2 - (r * cos(rad));
    y = tft.width/2 - (r * sin(rad));

    tft.drawLine(tft.width/2, tft.width/2, x, y, color2); //今のスピード
    ox=x;
    oy=y;
}

void drawScale(uint16_t color1, uint16_t color2) {
    int deg;
    float rad;
    double x0,y0,x1,y1;

    tft.drawCircle(tft.width/2,tft.width/2,R,color1);
    tft.fillRect(0, tft.width/2+1 , tft.width, 110, color2);

    for (deg=0; deg < 181; deg += 10) {

```

```

rad=PI * deg / 180; //convert deg to rad
x0 = tft.width/2 - ((R-3) * cos(rad));
y0 = tft.width/2 - ((R-3) * sin(rad));
x1 = tft.width/2 - ((R+2) * cos(rad));
y1 = tft.width/2 - ((R+2) * sin(rad));

tft.drawLine(x0, y0, x1, y1, GREY); //scale by 10km/h
}
}

void testdrawtext(char *text, uint16_t color) {
  tft.drawString(0, 0, text, color);
}

```

※-----※
 ※-----※

【注意事項】

TFT 液晶表示器のライブラリが提供されています。あらかじめダウンロードして、以下の手順でライブラリに追加しておきます。

1.TFT ライブラリのインストール

①.

<http://www.sainsmart.com/sainsmart-1-8-spi-lcd-module-with-microsd-led-backlight-for-arduino-mega-atmel-atmega.html>

のビデオの下にある

Documents

Download Link

をクリックして rar ファイルをダウンロードします。

②. rar を解凍し、TFT18 フォルダを

¥arduino-*.*.¥libraries or ¥Arduino の個人コード保存先¥libraries

の下にコピーします。

③. 次に Arduino IDE を起動すると、ライブラリが使えます。

ライブラリには、ヘッダファイルとライブラリソースファイルが含まれています。エディタで内容を確認し、ヘッダファイル名などについて確認しておきます。

プログラム上で、メーターの針、数値の描画前に黒で同じものを描いているのは、前回の表示を消している処理です。また、A/D の値が変化するときだけ描画するようにしています。このようにしておかなくとも、画面表示がちらついて見辛くなります。

※色は、液晶メーカーのサンプルのソースをそのまま使いました。RGB のデータの並びの R と B が入れ替わっているようですが、表示に問題はないのでそのままにしてあります。

9. 3. 8. 動作の様子

プログラムを Arduino に書込むと、リセットがかかり動き出します。初期化の後、図のような表示になります。

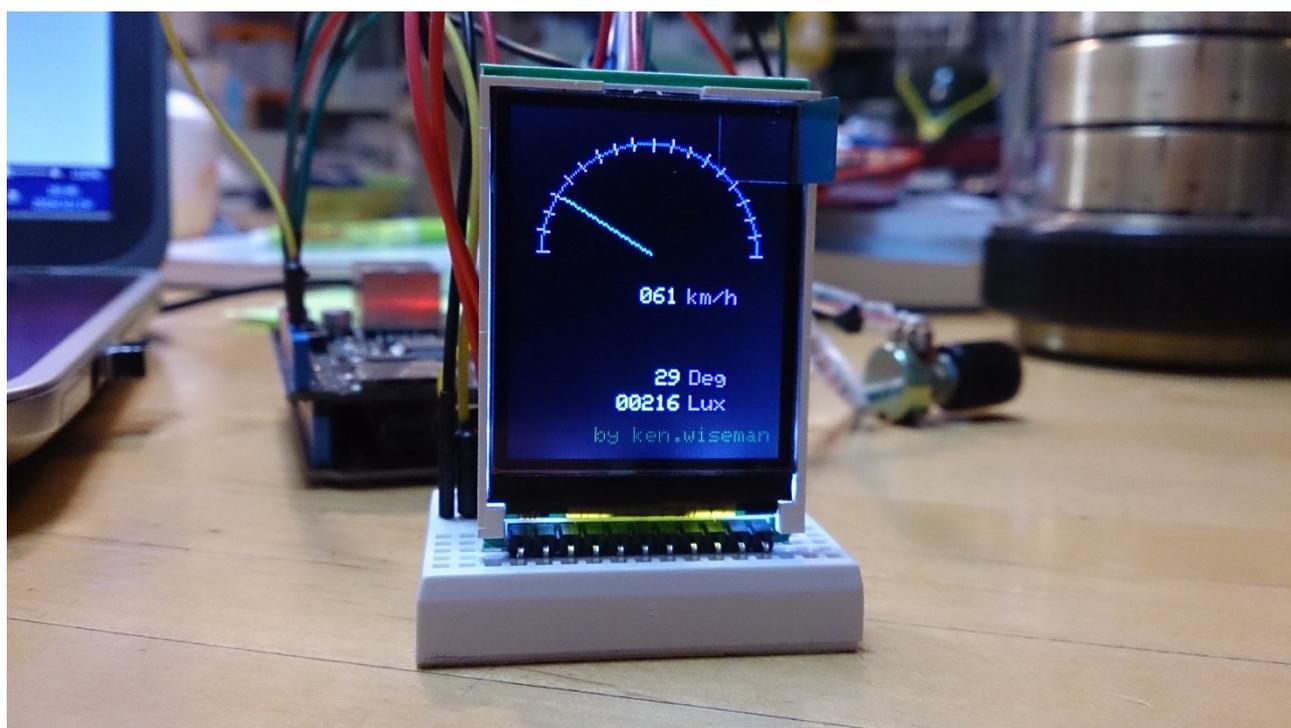


図 176 TFT 液晶によるデジタルメータ

A0 に接続されている VR を回すと、速度計に見立てたメーターの針が動き、速度表示が変化します。また、照度センサーの上に手をかざして影を作ると、Lux の値が減少します。温度センサーに指で触れると、温度が変化することが分かります。

温度センサー、照度センサーともにリニアな特性なので、校正は計算式の傾きの変更と定数を加える（または減じる）ことで行えます。

9. 3. 9. シリアル通信によるシステムの拡張

今回、作成したプログラムには、シリアル通信でスピードの値を送信する命令が含まれています。PC でシリアルポートからデータを受信し、この値を基に同じようなメーター表示機能を追加してみましょう。

そのためには、Processing と呼ばれるプログラミング環境を用意します。

Processing は、<https://processing.org/> から誰でもダウンロードできて自由に使うことができます。画面に Windows と Windows[Without Java]が表示されています。自分の PC に Java が入っているなら Windows[Without Java]を選択。Java が入っていないという人は通常の Windows を選択してダウンロードします。

ダウンロードしたファイルを解凍すればインストールが完了です。フォルダの中にある Processing.exe をダブルクリックして Processing を起動します。(下図)

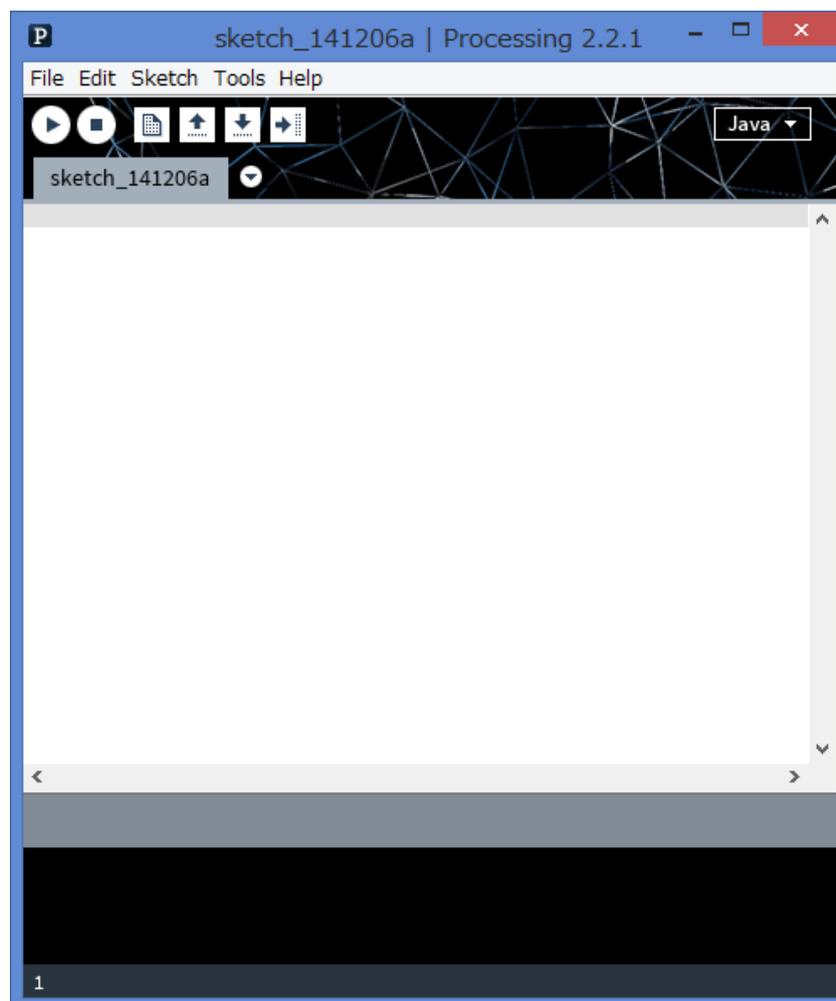


図 177 Processing 起動ウィンドウ

ここに、次のようなプログラムを書き込みます。このプログラムは、Java で書かれてい

ますが、難しいところは全くありません。最初の行でシリアル通信のモジュールを使っていることが分かるでしょう。また、プログラム全体は、前に作成したマイコン側のプログラムに驚くほど似ていることが分かります。

※-----※

※-----※

```
import processing.serial.*;
Serial port;

final int XX = 400;
final int YY = 400;
final int RR = 180;
final color BACK = #222211;
//final color NEEDLE = #FF7777;
final color NEEDLE = #99CCFF;
final color METER = #7799dd;
//final color SCALE = #AAAAAA;
final color SCALE = #FF7777;
final color SPEED = #EEEEEE;

int sp,osp;
int x,y,ox,oy;
PFont font;

void setup()
{

    font = loadFont("CenturyGothic-Italic-48.vlw");
    textFont(font);

    // キャンバスサイズ
    size(XX, YY); //canvas size
    //port=new Serial(this, "COM9", 9600);
    //port=new Serial(this, "COM11", 9600);
    port=new Serial(this, "COM14", 115200); //シリアルポート COM14 を
```

115200bps でオープンします。

```
    colorMode(RGB,256);
    background(BACK);
    SpeedMeter();
    osp=0;
    ox = int(width/2 - ((RR-8) * cos(0)));
    oy = int(width/2 - ((RR-8) * sin(0)));
    drawNeedle(0,0);

    drawSpeed(sp, osp, BACK, SPEED);

}

void draw()
{
    // background(#FFCC00); //set back ground color
    // ellipse(x, 100, 50, 50);
    if(sp!=osp) //受信したデータが、前回描画した時の値と異なる場合だけメータ
    を描画します。
    {
        drawNeedle(sp,osp);
        drawSpeed(sp, osp, BACK, SPEED);
        osp=sp;
    }
}

void drawSpeed(int sp, int osp, color c1, color c2)
{
    stroke(c1); //線の色
    fill(c1);
    rect(width/2-81,width/2+2,width/2+6,64);
    fill(c2);
    text(nf(sp,3)+" km/h ",width/2-80,width/2+60);
}
```

```
}
```

```
void SpeedMeter()
```

```
{
```

```
    stroke(METER); //線の色
```

```
    strokeWeight(1); //線の太さ
```

```
    noFill(); //塗りつぶさない
```

```
    arc(width/2, height/2, RR*2, RR*2, PI, TWO_PI);
```

```
    drawScale();
```

```
}
```

```
void drawNeedle(int sp, int osp) {
```

```
    float rad;
```

```
    int xx,yy;
```

```
    strokeWeight(3); //線の太さ
```

```
    stroke(BACK); //線の色
```

```
    line(width/2, width/2, ox, oy); //まず、以前の針を消す
```

```
    rad=PI * sp / 360.0; //scaling fucter = 0.5
```

```
    x = int(width/2 - ((RR-8) * cos(rad)));
```

```
    y = int(width/2 - ((RR-8) * sin(rad)));
```

```
    strokeWeight(1); //線の太さ
```

```
    stroke(NEEDLE); //線の色
```

```
    line(width/2, width/2, x, y); //今のスピード
```

```
    ox=x;
```

```
    oy=y;
```

```
}
```

```
void drawScale()
```

```
{
```

```
    int deg;
```

```
    float rad;
```

```
    int x0,y0,x1,y1;
```

```

for (deg=0; deg < 181; deg += 10) //速度 10km/h 単位に目盛を描く
{

    rad=PI * deg / 180; //convert deg to rad
    x0 = int(width/2 - ((RR-3) * cos(rad)));
    y0 = int(height/2 - ((RR-3) * sin(rad)));
    x1 = int(width/2 - ((RR+2) * cos(rad)));
    y1 = int(height/2 - ((RR+2) * sin(rad)));

    stroke(SCALE); //線の色
    line(x0, y0, x1, y1); //scale by 10km/h
}

}

//シリアル通信でマイコンからスピードの値を受信します。
void serialEvent(Serial p) //recieve from arduino
{
    sp=port.read();
}

```

※-----※
 ※-----※

Processing のプログラムも Arduino と同じように、`setup()`関数で、初期化を行います。

`draw()`関数でメイン処理を行います。今回のプログラムでは、シリアルポートからのデータ受信は、`serialEvent()`関数で処理しています。受信したデータは、`draw()`のメイン処理で、スピードメーターの針の位置として描画しています。

プログラムを書いたら、ウィンドウ左上の右向き三角印をクリックして、実行します。(下図)

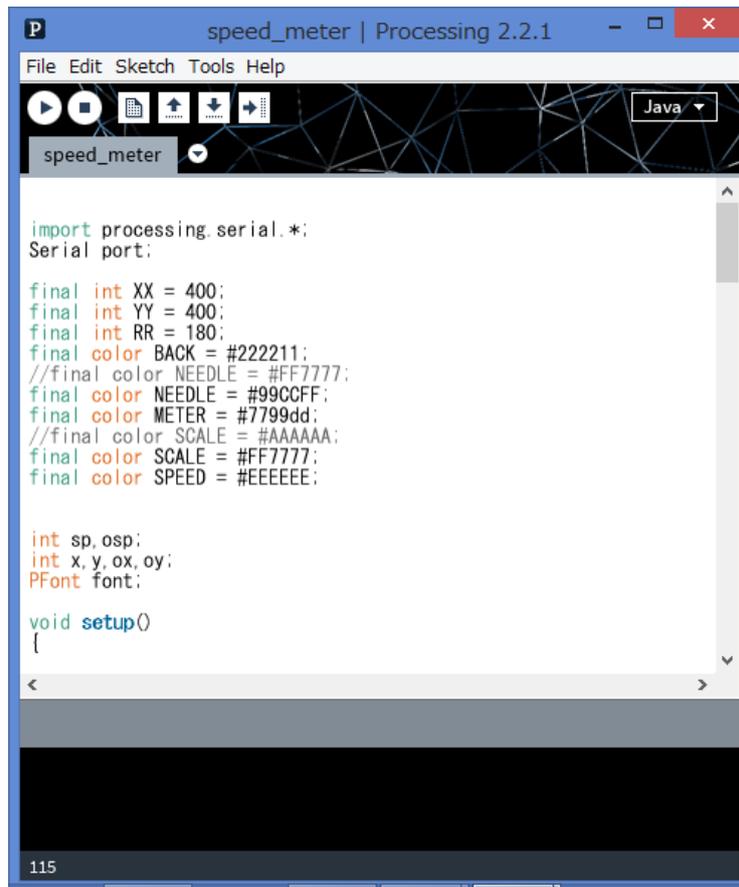


図 178 Processing のプログラム

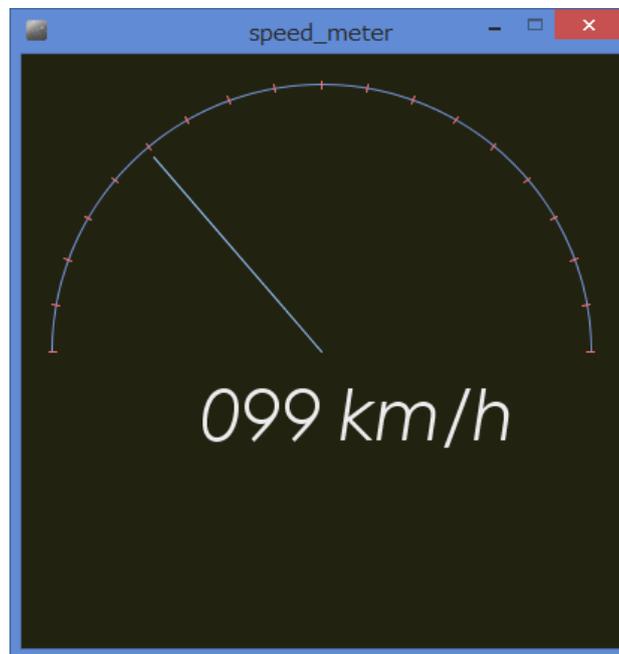


図 179 Processing の実行画面

別のウィンドウが開き、上図のようなスピードメーターが表示され、マイコンで表示している TFT グラフィック液晶のメーター表示と同じ値が表示されているでしょうか。ボリュームを回転すれば、TFT 表示と同期して PC 側の Processing のメーター表示も変化するはずです。

※文書の引用について

本書の内容は、次の各 HP からの引用があります。

記載された内容について下記の引用元が保証するものではありません。また、書かれている内容に不具合が出た場合でも JMAAB では一切の責任を負いかねます。

【引用元】

経済産業省	日本電気技術者協会
日本電池工業会	水素・燃料電池実証プロジェクト
東京ガス	DENSO(デンソー)
セキスイハイム	次世代自動車振興センター
大和総研	三菱自動車
東芝	新電元
国土交通省	Astamuze
太陽光発電協会	Wikipedia
産業技術総合研究所	ベクター (ベクタージャパン)
テキサスインスツルメンツ	monoist
日立ハイテック	特許流通報告
日本イーッテク	ホンダ
パワー社	科学技術振興機構
四国電力	国土交通省
LOOP	Arduino
燃料電池普及促進協会	Processing
燃料電池. net	

平成 26 年度文部科学省委託 「東日本大震災からの復興を担う専門人材育成支援事業」
東北の復興・再生を担う自動車組込みエンジニア育成支援プロジェクト

■推進協議会

- ◎ 佐藤 公一 東北電子専門学校 校長
今野 幸信 東北電子専門学校 総務部 部長
與那嶺 尚弘 仙台高等専門学校 知能エレクトロニクス工学科 准教授
岩渕 喜悦 岩淵技術事務所 代表
町田 尚彦 宮城県黒川高等学校 教頭
木村 康弘 宮城県米谷工業高等学校 情報技術科 科長
白田 正樹 株式会社コー・ワークス システム開発事業部
渋谷 義博 トライポッドワークス株式会社 先端技術開発部 部長
三方 雅仁 創造技研株式会社 代表取締役
羽曾部 恭美 カストマシステム株式会社
エンベデソドシステムソリューション事業部 事業部長
須藤 敬行 宮城県経済商工観光部 産業人材対策課 課長補佐(班長)
本圖 雅也 宮城県経済商工観光部 自動車産業振興室 技術支援班 主事
今井 和彦 宮城県震災復興・企画部 情報産業振興室/
産業技術総合センター 機械電子情報技術部 情報技術開発班
吉岡 正勝 有限会社ザ・ライスマウンド マーケティングマネージャー

■開発分科会

- ◎ 坂藤 健 東北電子専門学校 組込みソフトエンジニア科 学科主任
高橋 敬 東北電子専門学校 CAD設計製図科 学科主任
阿保 隆徳 東北電子専門学校 デジタル家電科 学科主任
白田 正樹 株式会社コー・ワークス システム開発事業部
羽曾部 恭美 カストマシステム株式会社
エンベデソドシステムソリューション事業部 事業部長
柴原 健次 エキスパートプロモーション 代表
原田 賢一 有限会社ワイズマン 代表取締役
吉岡 正勝 有限会社ザ・ライスマウンド マーケティングマネージャー

■講座運営分科会

- ◎ 坂藤 健 東北電子専門学校 組込みソフトエンジニア科 学科主任
高橋 敬 東北電子専門学校 CAD設計製図科 学科主任
阿保 隆徳 東北電子専門学校 デジタル家電科 学科主任
小野寺 敬司 花壇自動車大学校 教頭
木村 康弘 宮城県米谷工業高等学校 情報技術科 科長
三方 雅仁 創造技研株式会社 代表取締役
羽曾部 恭美 カストマシステム株式会社
エンベデソドシステムソリューション事業部 事業部長
春木 賢仁 株式会社アフレル
吉岡 正勝 有限会社ザ・ライスマウンド マーケティングマネージャー

平成 26 年度文部科学省委託 「東日本大震災からの復興を担う専門人材育成支援事業」
東北の復興・再生を担う自動車組込みエンジニア育成支援プロジェクト

次世代自動車基礎教材

平成 27 年 2 月

東北の復興・再生を担う自動車組込みエンジニア育成支援プロジェクト推進協議会

学校法人日本コンピュータ学園（東北電子専門学校）
〒980-0013 宮城県仙台市青葉区花京院一丁目3番1号

●本書の内容を無断で転記、掲載することは禁じます。