

## 24. 組み込みシステムに関する知識Ⅱ

### 1. 科目の概要

組み込みシステムの開発について、その開発手順や開発プロジェクト管理、体制などについて解説する。また組み込みシステム自体のアーキテクチャや組み込みシステムを取り巻くネットワーク環境の特徴やソフトウェア/ハードウェア要件などについて解説する。

### 2. 習得ポイント

本科目の学習により習得することが期待されるポイントは以下の通り。

習得ポイント	説明	シラバスの対応コマ
II-24-1. 組み込みシステムの開発技術	組み込みシステムの開発について、システム全体の開発工程やハードウェア、ソフトウェア開発の考え方、開発ライフサイクルや開発プロセスの概要、管理方法など、システム開発の基本的な考え方や主要な開発技法、特徴や留意点を解説する。	9
II-24-2. 組み込みシステム開発手順と関連文書	組み込みシステム開発工程の各ステップで利用できる各種の技法を説明し、ハードウェア構成図、ソフトウェア構成図、制御データフロー図やコントロールフロー図など、それぞれの手順で生成される成果物の概要と位置付けを解説する。	9
II-24-3. オブジェクト指向分析による組み込みシステム開発	制御対象やシステム資源をオブジェクトとして表現して設計するオブジェクト指向分析を、組み込み開発へ適用する方法について説明する。状態遷移図による分析や入出力の関係を表示する方法、リアルタイム性や事象応答精度の表現などオブジェクト指向によるモデリングを解説する。	10
II-24-4. 組み込みシステム開発における各種の管理	ハードウェアやソフトウェア個別の品質特性や開発における品質維持など組み込みシステムの品質管理や、特許・ライセンス・機密情報などの管理、開発したシステムの変更管理など、組み込みシステム開発で必要となる各種の管理について説明する。	11
II-24-5. 組み込みシステム開発プロジェクトの体制	組み込みシステム開発プロジェクトにおけるプロジェクトマネジメントに必要な各種のマネジメント、スケジュールマネジメント、リソースマネジメント、要求マネジメント、コミュニケーションマネジメント、について解説を行うと共に、組み込みシステム開発プロジェクトをマネジメントする鍵を提示する。	11
II-24-6. 組み込みシステムのアーキテクチャ	PC/AT互換機や情報家電、携帯電話、センサシステムなど代表的な組み込みシステムのハードウェアを紹介する。またこれらのシステムにおいて動作するソフトウェア構成の代表例を示し、それぞれの特徴や開発に必要な項目について説明する。	12
II-24-7. 組み込みシステムにおけるネットワークのハードウェア要件	シリアル通信、赤外線インタフェース、IEEE802.15などの低速デバイスや、IEEE1394、イーサネット、USB2.0などの高速デバイスによる通信インタフェースのハードウェアについて、その概要や特徴、利用方法、留意点を解説する。	13
II-24-8. 組み込みシステムにおけるネットワークのソフトウェア	組み込みシステムで利用できる各種の通信デバイスに関して、ソフトウェアからの利用方法を説明する。また通信するデータや通信対象、通信手段の選択方法や、組み込みシステムで利用できる通信ミドルウェアについて解説する。	13
II-24-9. 組み込みコンピュータを取り巻く今後の環境	携帯電話、PDAのプラットフォームや情報家電など、今後発展が期待される組み込みシステム応用分野の概要や将来展望、それらの現状と発展の動向など、組み込みシステムを取り巻く様々な環境の変化について解説する。	15
II-24-10. インターネット以外の組み込みコンピュータネットワーク	携帯電話による通信や、赤外線通信、コピキタネットワークなど、組み込みシステムで利用できるインターネット以外のネットワークアーキテクチャについて解説する。さらに組み込みシステムにおけるIPv6の利用動向についても述べる。	15

#### 【学習ガイダンスの使い方】

- 「習得ポイント」により、当該科目で習得することが期待される概念・知識の全体像を把握する。
- 「シラバス」、「IT 知識体系との対応関係」、「OSS モデルカリキュラム固有知識」をもとに、必要に応じて、従来の IT 教育プログラム等との相違を把握した上で、具体的な講義計画を考案する。
- 習得ポイント毎の「学習の要点」と「解説」を参考にして、講義で使用する教材等を準備する。

### 3. IT 知識体系との対応関係

「24. 組み込みシステムに関する知識Ⅱ」と IT 知識体系との対応関係は以下の通り。

科目名	基本レベル(I)																応用レベル(II)															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
24. 組み込みシステムに関する知識Ⅱ	<組み込みシステムとは何か>	<組み込みコンピュータのアーキテクチャ>	<組み込みシステムの基本構成>	<組み込みコンピュータハードウェアの基本>	<CPUアーキテクチャの基本>	<組み込みソフトウェアの概要>	<カーネル処理の概要>	<リアルタイムシステムの構成と仕組み>	<組み込みシステムの開発方法>	<オブジェクト指向分析による組み込みシステムの分析とモデリング>	<組み込みコンピュータシステムの開発管理>	<組み込みシステムのアーキテクチャ>	<組み込みシステムのネットワーク機能>	<組み込みシステムの設計>	<組み込みシステムの動作>	<これからの組み込みシステムの動向>																

[シラバス : [http://www.ipa.go.jp/software/open/oss/download/Model\\_Curriculum\\_05\\_24.pdf](http://www.ipa.go.jp/software/open/oss/download/Model_Curriculum_05_24.pdf)]

#### <IT 知識体系上の関連部分>

分野	科目名	基本レベル(I)													応用レベル(II)																
		1	2	3	4	5	6	7	8	9	10	11	12	13	1	2	3	4	5	6	7	8	9	10	11	12	13				
組織関連事項と情報システム	1	IT-IAS1 情報セキュリティ	IT-IAS2 情報セキュリティの性	IT-IAS3 運用上の問題	IT-IAS4 ホリ	IT-IAS5 攻撃	IT-IAS6 情報セキュリティ分野	IT-IAS7 フォレンジック(情報証拠)	IT-IAS8 情報の収集	IT-IAS9 情報セキュリティポリシー	IT-IAS10 脅威分析モデル	IT-IAS11 脆弱性																			
	2	IT-SP 社会的な観点とフェデラルとしての課題	IT-SP1 プロフェッショナルとしてのコミュニケーション	IT-SP2 コンピュータの歴史	IT-SP3 コンピュータを取り巻く社会環境	IT-SP4 チームワーク	IT-SP5 知的財産権	IT-SP6 コンピュータの法的問題	IT-SP7 組織の中でのIT	IT-SP8 プロフェッショナルとしての倫理的な問題と責任	IT-SP9 プライバシーと個人の自由																				
応用技術	3	IT-IM 情報管理	IT-IM1 情報管理の概念と基礎	IT-IM2 データベース関係性	IT-IM3 データアーキテクチャ	IT-IM4 データモデリングとデータベース設計	IT-IM5 データと情報の管理	IT-IM6 データベースの応用分野																							
	4	IT-WS Webシステムとその技術	IT-WS1 Web技術	IT-WS2 情報アーキテクチャ	IT-WS3 デジタルメディア	IT-WS4 Web開発	IT-WS5 脆弱性	IT-WS6 ソーシャルソフトウェア																							
ソフトウェアの方法と技術	5	IT-PF プログラミング基礎	IT-PF1 基本データ構造	IT-PF2 フログラミングの基本的構成要素	IT-PF3 オブジェクト指向プログラミング	IT-PF4 アルゴリズムと問題解決	IT-PF5 イベント駆動プログラミング	IT-PF6 再帰																							
	6	IT-PT 技術を統合するためのプログラミング	IT-PT1 システム間連携	IT-PT2 データやり取りと交換	IT-PT3 統合的コーディング	IT-PT4 スクリプティング手法	IT-PT5 ソフトウェアセキュリティの実現	IT-PT6 種々の問題	IT-PT7 プログラミング言語の概要																						
	7	IT-SE ソフトウェア工学	IT-SE1 歴史と概要	IT-SE2 ソフトウェアプロセス	IT-SE3 ソフトウェアの要求と仕様	IT-SE4 ソフトウェアの設計	IT-SE5 ソフトウェアのテストと検証	IT-SE6 ソフトウェアの開発・保守ツールと環境	IT-SE7 ソフトウェアプロジェクト管理	IT-SE8 言語翻訳	IT-SE9 ソフトウェアのフォールトトレランス																				
	8	IT-SIA システムインテグレーションとアーキテクチャ	IT-SIA1 要求仕	IT-SIA2 調達/手配	IT-SIA3 インテグレーション	IT-SIA4 プロジェクト管理	IT-SIA5 テストと品質保証	IT-SIA6 組織の特性	IT-SIA7 アーキテクチャ																						
システム構築	9	IT-NET ネットワーク	IT-NET1 ネットワークの基礎	IT-NET2 ルーティングとスイッチング	IT-NET3 物理層	IT-NET4 セキュリティ	IT-NET5 アプリケーション分野	IT-NET6 ネットワーク管理																							
	10	IT-NIK テレコミュニケーション	IT-NIK1 歴史と概要	IT-NIK2 通信ネットワークのアーキテクチャ	IT-NIK3 通信ネットワークのプロトコル	IT-NIK4 LANとWAN	IT-NIK5 クラウドサービスとセキュリティ	IT-NIK6 データのセキュリティと整合性	IT-NIK7 データ通信	IT-NIK8 ファイアウォールとセキュリティ	IT-NIK9 無線ネットワーク	IT-NIK10 性能評価	IT-NIK11 ネットワーク管理	IT-NIK12 圧縮と伸張																	
	11	IT-PI プラットフォーム技術	IT-PI1 オペレーティングシステム	IT-PI2 アーキテクチャと機構	IT-PI3 コンピュータインフラストラクチャ	IT-PI4 デバイスドライバソフトウェア	IT-PI5 ファームウェア	IT-PI6 ハードウェア																							
コンピュータハードウェア	12	IT-OPS オペレーティングシステム	IT-OPS1 歴史と概要	IT-OPS2 実行性	IT-OPS3 スケジューリングとデスマッチ	IT-OPS4 メモリ管理	IT-OPS5 セキュリティと保護	IT-OPS6 ファイル管理	IT-OPS7 リアルタイムOS	IT-OPS8 OSの概要	IT-OPS9 OSの設計の原則	IT-OPS10 デバイスマネジメント	IT-OPS11 システム性能評価																		
	13	IT-CAO コンピュータアーキテクチャと構成	IT-CAO1 歴史と概要	IT-CAO2 コンピュータアーキテクチャの進化	IT-CAO3 メモリシステムの構成とアーキテクチャ	IT-CAO4 インタフェースと通信	IT-CAO5 ハイパフォーマンスシステム	IT-CAO6 CPUアーキテクチャ	IT-CAO7 性能・コスト評価	IT-CAO8 分散・並列処理	IT-CAO9 分散コンピュータによる計算	IT-CAO10 コンピュータによる計算	IT-CAO11 ネットワーク	IT-CAO12 性能評価																	
複数環境にまたがるもの	14	IT-ITF IT基礎	IT-ITF1 ITの歴史的なテーマ	IT-ITF2 組織の問題	IT-ITF3 ITの歴史	IT-ITF4 IT分野(学際)とそれに関連する分野(学際)	IT-ITF5 応用領域	IT-ITF6 IT分野における数学と統計学の活用																							
	15	IT-ESI 組み込みシステム	IT-ESI10 歴史と概要	IT-ESI11 高電力コンピュータアーキテクチャ	IT-ESI12 高信頼性システムの設計	IT-ESI13 組み込みアーキテクチャ	IT-ESI14 開発環境	IT-ESI15 ライフサイクル	IT-ESI16 要件分析	IT-ESI17 仕様定義	IT-ESI18 構造設計	IT-ESI19 テスト	IT-ESI20 プロジェクト管理	IT-ESI21 並行設計(ハードウェア、ソフトウェア)	IT-ESI22 実装																

## 4. OSS モデルカリキュラム固有の知識

OSS モデルカリキュラム固有の知識は、OSS を用いた組み込みシステムに関する知識である。OSS を活用した携帯電話などの事例を扱う。

科目名	第9回	第10回	第11回	第12回	第13回	第14回	第15回
24.組み込みシステムに関する知識Ⅱ	(1)組み込み型コンピュータ応用システムの開発技術 (2)開発方法と成果物	(1)オブジェクト指向のアプローチ (2)設計の評価と改善	(1)品質管理 (2)組み込みシステム開発のプロジェクト特 (3)機密・契約管理 (4)変更管理	(1)PC/ATアーキテクチャ (2)情報家電 (3)携帯電話のアーキテクチャ (4)センサーシステム(単一の機能で構成されるシステ	(1)通信ハードウェア (2)低速な通信インタフェース (3)高速・大容量な通信インタフェース (4)通信ミドルウェア (5)組み込みシステムにおける通信機能の設計	(1)業務要件からのシステム全体設計 (2)ソフトウェア設計 (3)組み込みシステムとしての評価 (4)全体の性能、拡張性、コスト要件などを評価	(1)非組み込みコンピュータによる基盤 (2)非インターネット

(網掛け部分は IT 知識体系で学習できる知識を示し、それ以外は OSS モデルカリキュラム固有の知識を示している)

スキル区分	OSS モデルカリキュラムの科目	レベル
組み込み分野	24. 組み込みシステムに関する知識Ⅱ	応用
習得ポイント	Ⅱ-24-1. 組み込みシステムの開発技術	
対応する コースウェア	第9回（組み込みシステムの開発方法）	

## Ⅱ-24-1. 組み込みシステムの開発技術

組み込みシステムの開発について、システム全体の開発工程やハードウェア、ソフトウェア開発の考え方、開発ライフサイクルや開発プロセスの概要、管理方法など、システム開発の基本的な考え方や主要な開発技法、特徴や留意点を解説する。

### 【学習の要点】

- \* 組み込みシステムの開発は多様で複雑である。
- \* 必要な機能と要素技術を決定する工程がシステムエンジニアリングプロセスである。
- \* システムエンジニアリングプロセスは大きく分けて、システム要求分析とシステムアーキテクチャ設計に分けられる。
- \* システム要求分析では、システム全体の機能仕様と制約を分析し、定義する。
- \* システムアーキテクチャ設計はシステム要求分析の後に行われ、システム全体のアーキテクチャを決定する。
- \* システムエンジニアリングプロセスにより、ハードウェア・ソフトウェアの機能実装についての仕様が明らかになるので、ハードウェア設計とソフトウェア設計に移行することができる。

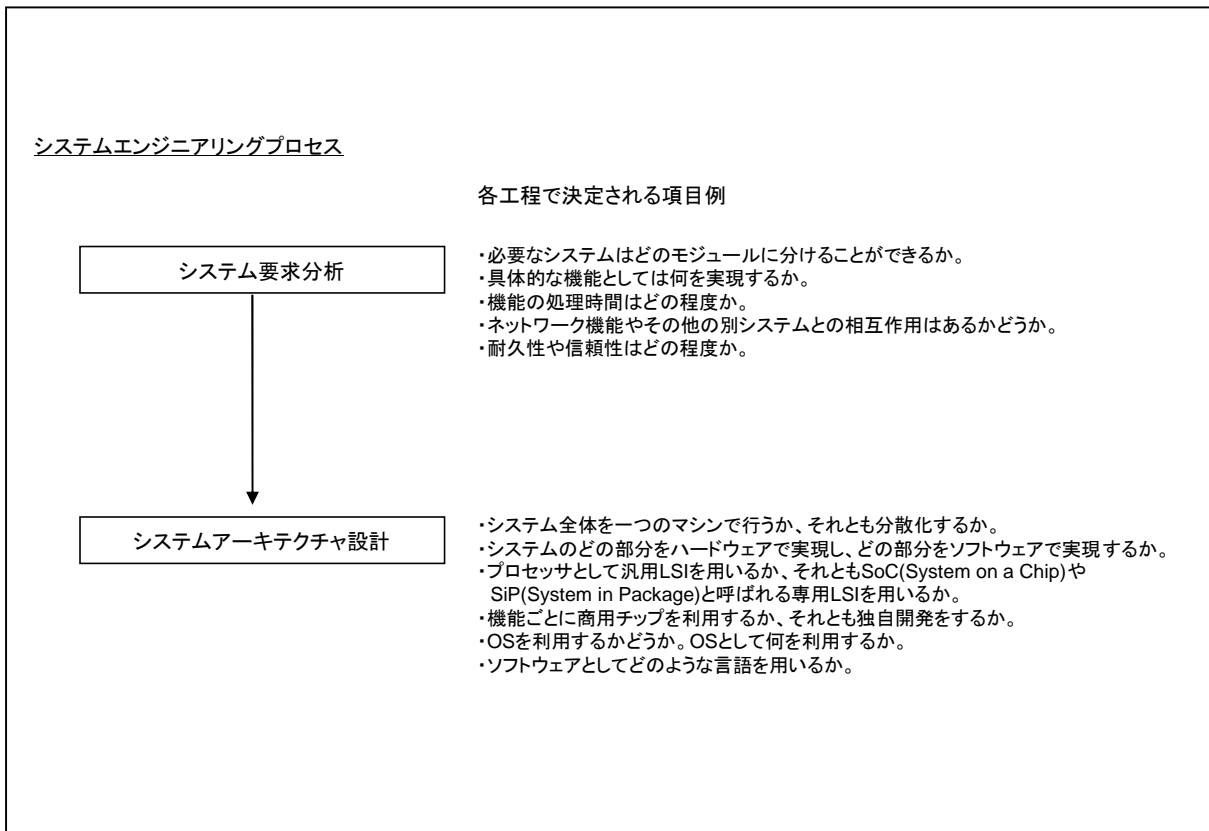


図 Ⅱ-24-1. 組み込みシステムの開発技術

## 【解説】

### 1) 組み込みシステムの開発

- \* 組み込みシステムの開発と一口でいってもその内容は多様性に富んでいる。例えばリアルタイム OS についても 100 種類を超える選択肢があり、実装に必要な要素技術についてもソフトウェア、ハードウェア共にあまりに多様である。従って組み込みシステムの開発については、仕様と要件に対してどのような要素技術が必要であるかを判断することが必要になる。
- \* 例えば携帯電話の開発について言えば、通話に必要なベースバンド処理、音声の符号・復号処理、音楽再生処理、カメラ撮影・画像処理等の多様な機能を実現することが必要でそのための個々の要素技術をどう利用するかという判断が必要である。
- \* 組み込みシステムの開発において、ハードウェアに必要な技術・ソフトウェアに必要な技術を判断していく工程をシステムエンジニアリングプロセスという。システムエンジニアリングプロセスは大きく分けて、システム要求分析とシステムアーキテクチャ設計に分けることができる。

### 2) システム要求分析

- \* システム要求分析とはシステムエンジニアリングプロセスの最初の工程にあたるものである。ここではシステム全体の機能仕様と制約を定義し分析する。
- \* 具体的にはシステムにどのような機能が要求されているか、システムに対する制約条件(コスト、処理時間、物理的サイズ、バッテリー等の外部要件)を明らかにし、必要な機能を確定することが必要である。
- \* システム要求分析で行われるのは具体的には次のような決定である。
  - 必要なシステムはどのモジュールに分けることができるか。
  - 具体的な機能としては何を実現するか。
  - 機能の処理時間はどの程度か。
  - ネットワーク機能やその他の別システムとの相互作用はあるかどうか。
  - 耐久性や信頼性はどの程度か。
- \* これらの作業に必要な人材リソースやハードウェア・ソフトウェアに関する要素技術の洗い出しを行うこともある。

### 3) システムアーキテクチャ設計

- \* システムアーキテクチャ設計は、システム要求分析の後に行われる工程で、システム全体のアーキテクチャを決定することを目的とする。
- \* システムアーキテクチャ設計で具体的に行われるものは次のような決定である。
  - システム全体を一つのマシンで行うか、それとも分散化するか。
  - システムのどの部分をハードウェアで実現し、どの部分をソフトウェアで実現するか。
  - プロセッサとして汎用 LSI を用いるか、それとも SoC(System on a Chip)や SiP(System in Package)と呼ばれる専用 LSI を用いるか。
  - 機能ごとに商用チップを利用するか、それとも独自開発をするか。
  - OS を利用するかどうか。OS として何を利用するか。
  - ソフトウェアとしてどのような言語を用いるか。
- \* システムアーキテクチャ設計が終了すれば、次はハードウェア・ソフトウェア設計に移行する。

スキル区分	OSS モデルカリキュラムの科目	レベル
組み込み分野	24. 組み込みシステムに関する知識Ⅱ	応用
習得ポイント	Ⅱ-24-2. 組み込みシステム開発手順と関連文書	
対応する コースウェア	第9回（組み込みシステムの開発方法）	

## Ⅱ-24-2. 組み込みシステム開発手順と関連文書

組み込みシステム開発工程の各ステップで利用できる各種の技法を説明し、ハードウェア構成図、ソフトウェア構成図、制御データフロー図やコントロールフロー図など、それぞれの手順で生成される成果物の概要と位置付けを解説する。

### 【学習の要点】

- \* 機器が高機能化・複雑化するに伴い、複数の開発者間での意思疎通のためにも組み込みシステム開発工程の各ステップにおいてドキュメントの重要性が高まっている。
- \* 各工程での成果物としては、ハードウェア構成図、ソフトウェア構成図、制御データフロー図やコントロールフロー図などがある。
- \* ソフトウェアの開発工程であるソフトウェアプロセスを例にとるとにはウォーターフローモデル、V字モデル、プロトタイプングモデル、アジャイルプログラミングモデルなどの開発手法がある。

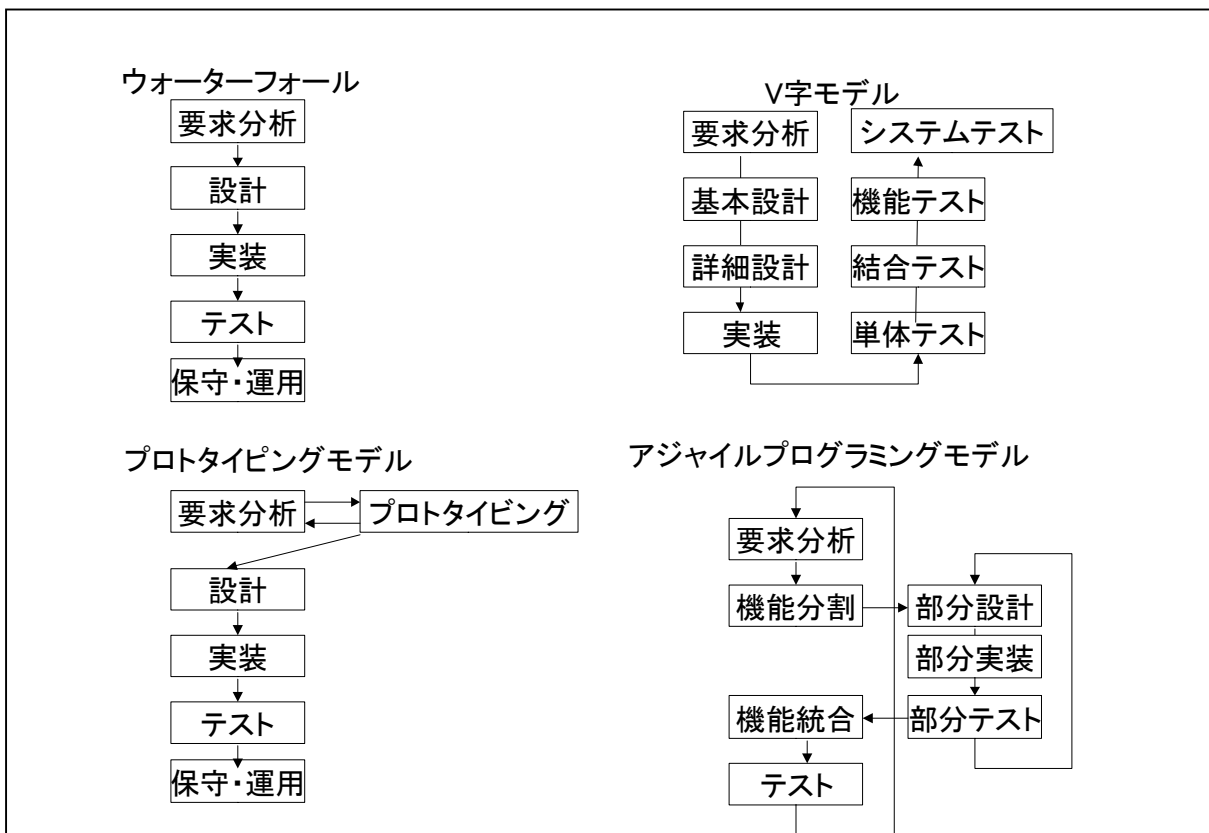


図 Ⅱ-24-2. 様々な組み込みシステムの開発プロセス

## 【解説】

### 1) 組み込みシステム開発工程と成果物

- \* 組み込みシステム開発における工程は、ハードウェア設計、ハードウェア実装、ソフトウェア設計、ソフトウェア実装などに分けられる。
- \* 各工程において担当者間の意思疎通、次工程への的確な情報伝達、および品質テストや継続的な機能改善のための資料としてドキュメントを残すことが重要である。
- \* 主なドキュメントには下記がある。
  - ハードウェア構成図: 基板設計書や利用する部品一覧など
  - ソフトウェア構成図: 利用言語、機能構成図など
  - 制御データフロー図: データと処理の流れをまとめたものなど
  - コントロールフロー図: イベントごとのプロセスの起動や停止フローをまとめたものなど

### 2) 開発手順例

- \* 開発手順の例として、組み込み開発でのソフトウェアプロダクトを開発するために行われる一連の作業や活動であるソフトウェアプロセスを取り上げる。
- \* ソフトウェアの開発は過度に知的労働集約的な作業であるために、このプロセスを効率的にかつ創造的に管理する技法によって、ソフトウェアプロダクトの質と開発速度に違いが出る。
- \* 組み込みソフトウェアプロセスには複数のプロセスモデルがある。
  - ウォーターフォールモデル: ウォーターフォールモデルは、要求分析 → 設計 → 実装 → テスト → 保守・運用 という一連のプロセスが一つの流れに沿って進むモデルである。
  - V字モデル: V字モデルは、ウォーターフォールモデルにおけるテストを四つのレベル、単体テスト、統合テスト、機能テスト、システムテストに分けてとらえ、後半の各段階を前半の実装、詳細設計、基本設計、要求分解の段階に対応させたモデルである。
  - プロトタイプモデル: プロトタイプモデルは、要求分析の段階でプロトタイプ制作を行いその検証をもって、設計 → 実装 → テスト → 保守・運用につなげるモデルで、要求分析段階における要求の曖昧さを減らすことを重視している。
  - アジャイルプログラミングモデル: アジャイルプログラミングモデルは、インクリメンタル型プロセスモデルともいわれ、開発対象をモジュール単位、優先度単位、要求の明確な単位に分け、これらの部分開発を繰り返して漸次的に開発を進めるモデルである。要求の決定に時間がかからないようにすることを重視している。

スキル区分	OSS モデルカリキュラムの科目	レベル
組み込み分野	24. 組み込みシステムに関する知識Ⅱ	応用
習得ポイント	Ⅱ-24-3. オブジェクト指向分析による組み込みシステム開発	
対応する コースウェア	第 10 回 (オブジェクト指向分析による組み込みシステムの分析とモデリング)	

## Ⅱ-24-3. オブジェクト指向分析による組み込みシステム開発

制御対象やシステム資源をオブジェクトとして表現して設計するオブジェクト指向分析を、組み込み開発へ適用する方法について説明する。状態遷移図による分析や入出力の関係を表現する方法、リアルタイム性や事象応答精度の表現などオブジェクト指向によるモデリングを解説する。

### 【学習の要点】

- \* 組み込みシステムのソフトウェア開発において、オブジェクト指向分析をすることがある。
- \* ソフトウェアの要求分析と設計にはソフトウェアモデリングのパラダイムが必要である。
- \* ソフトウェアモデリングには、オブジェクト指向の他に、構造化手法、アスペクト指向等もある。

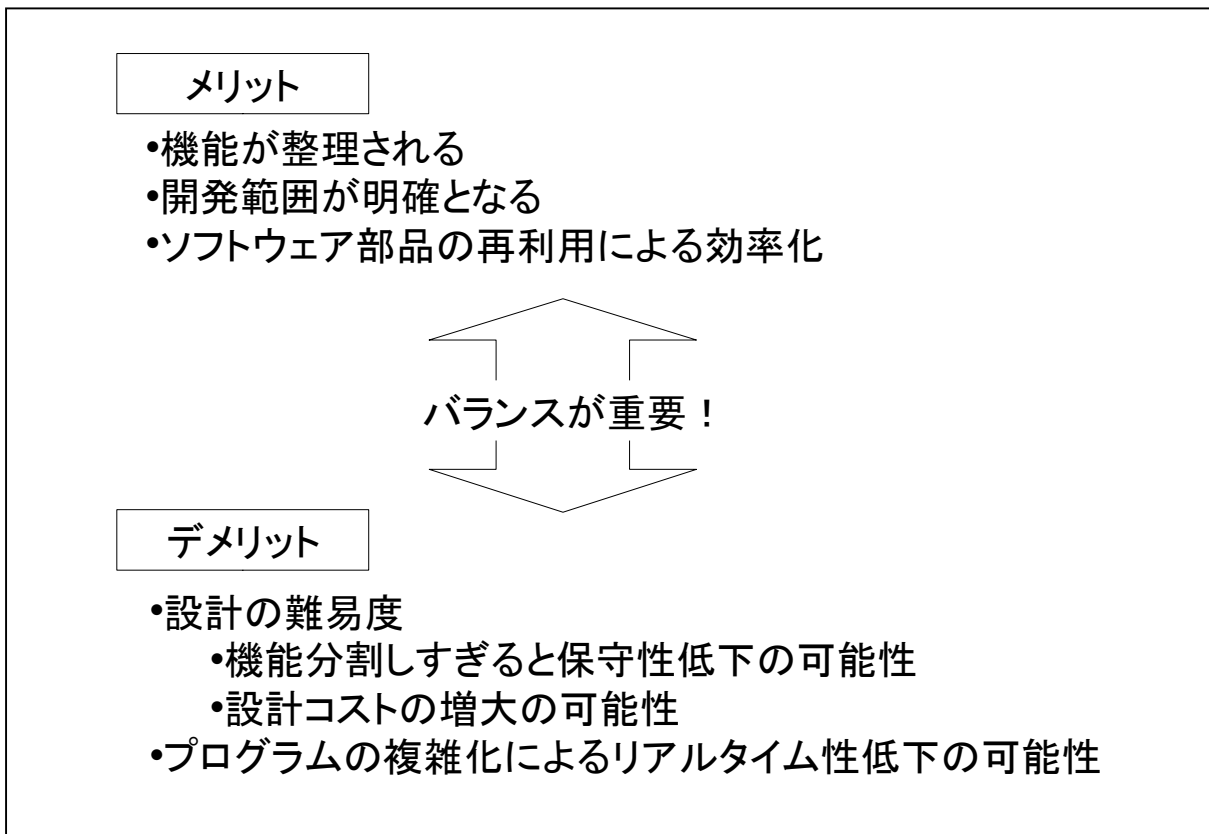


図 Ⅱ-24-3. 組み込み開発におけるオブジェクト指向分析のメリット・デメリット



## 【解説】

### 1) オブジェクト指向分析

- \* オブジェクトを主要な構成要素としてソフトウェアを構築する手法である。(II-13-1 参照)
- \* UML 等の記法を利用して、オブジェクトを抽象化し、そのオブジェクト間の相互作用を定義する設計手法である。
- \* 状態遷移図による分析や入出力の関係を表現することがある。

### 2) オブジェクト指向分析による組み込み開発の特徴と留意点

- \* オブジェクト指向分析により、オブジェクトとしての機能やドメインとしての開発範囲などが明確になる。
- \* UML やユースケース図、データフロー図により複雑な処理を必要とするシステムにおいても設計段階で要件を的確に整理できる。
- \* ソフトウェアの再利用を進めやすくなり開発効率の改善に役立てられる。
- \* 開発効率をあげるためにハードウェアに依存する機能(センサなど)とソフトウェアを切り離したアプローチが有用な場合がある。
- \* オブジェクト指向分析に対応した組み込み開発環境が必要となる。
- \* 設計の難易度が上がり、機能分割を進めてオブジェクト数を増やしすぎた場合、構造が複雑になり保守性の低下を招く可能性がある。また設計コストの増大にも配慮が必要である。
- \* プログラムの規模は大きくなるため、リアルタイム性への配慮も重要となる。

### 3) ソフトウェアモデリングと抽象化

- \* ソフトウェアの要求分析と設計においては、問題ドメインを効率良く抽象化し、モデリングをするためのモデリングパラダイムが必要である。
- \* オブジェクト指向以外のソフトウェアモデリングのパラダイムについては以下のものがある。
  - 構造化手法:機能に注目し、機能ごとに段階的にモデリングする手法である。データフローダイアグラム等を利用してモデリングを行う。
  - アスペクト指向  
構造化手法もオブジェクト指向も一般に、モジュール分割する手法であるが、そのために各モジュールに要素が分割してしまうという欠点がある。このことが保守性の低下をもたらす。これを解決するために「アスペクト」という概念で、モジュールに分割された要素を横断的にカバーするのがアスペクト指向手法である。
- \* 保守性と再利用性のために、ソフトウェアコンポーネントとデザインパターンを流用する。

スキル区分	OSS モデルカリキュラムの科目	レベル
組み込み分野	24. 組み込みシステムに関する知識Ⅱ	応用
習得ポイント	Ⅱ-24-4. 組み込みシステム開発における各種の管理	
対応する コースウェア	第 11 回（組み込みコンピュータシステムの開発管理）	

## Ⅱ-24-4. 組み込みシステム開発における各種の管理

ハードウェアやソフトウェア個別の品質特性や開発における品質維持など組み込みシステムの品質管理や、特許・ライセンス・機密情報などの管理、開発したシステムの変更管理など、組み込みシステム開発で必要となる各種の管理について説明する。

### 【学習の要点】

- \* 組み込みシステムには非常に高い品質が求められる。これを担保するための最も重要な仕組みがテストである。
- \* 自社の特許を技術ライセンスと提供するビジネスも盛んに行われている。
- \* 機器の高性能化に伴い、開発規模の拡大による複数人による開発、または継続的な開発と改良、品質保証のためには、ハードウェア、ソフトウェア共に、構成管理をおこなう必要がある。

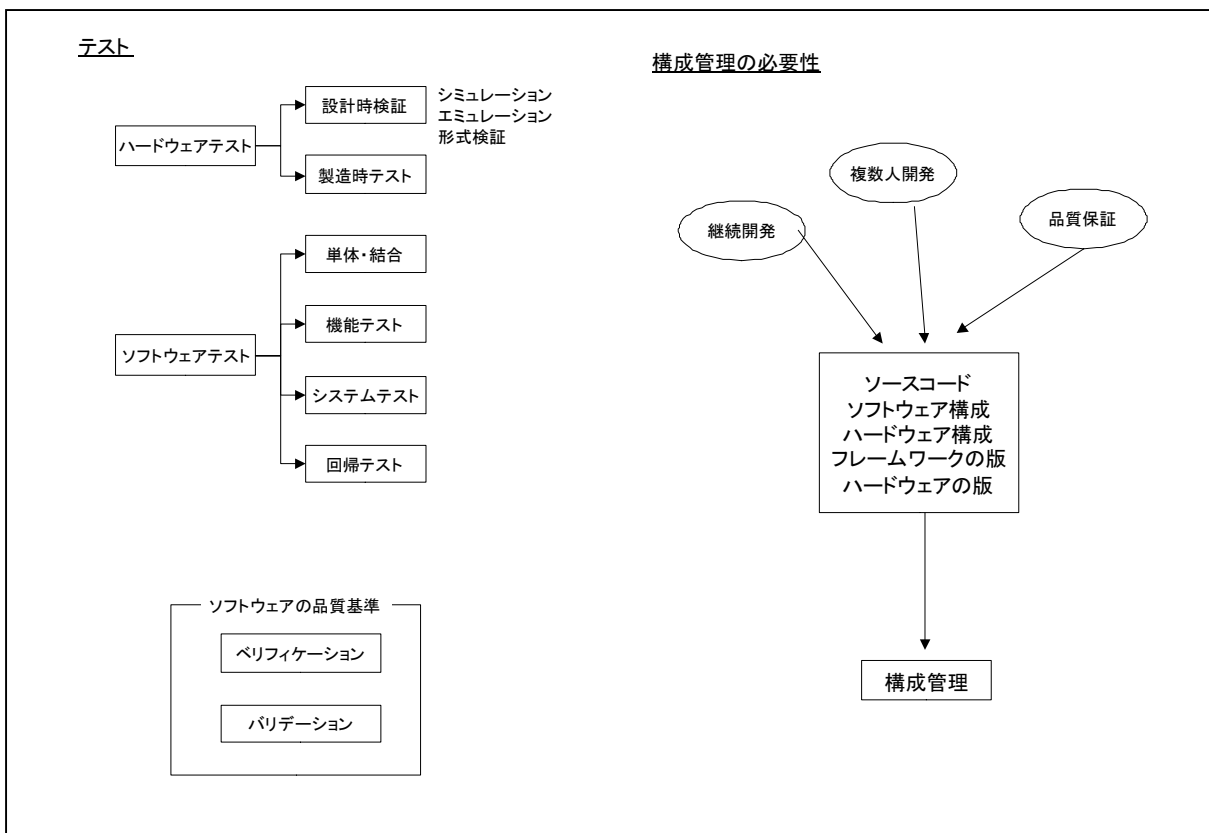


図 Ⅱ-24-4. 組み込みにおけるテストと構成管理

## 【解説】

### 1) 品質管理

- \* 組み込みシステムは、一度稼働させた製品は止まることなく数年または数十年に渡って誤動作なく稼働するという、非常に高い品質が求められている。そのために、必要な品質管理が大変重要である。品質管理においては、最も重要なのはテストである。
- \* ハードウェアにおいては、設計段階において行われる検証と、製造段階で行われるテストに分けられる。
  - 設計時検証:例えば SoC において設計が期待通りに行われているかどうかをチェックするためには、シミュレーションによる検証、エミュレーションによる検証、形式的な検証などが必要である。
  - 製造時テスト:ハードウェア設計が成功したとしても、製造過程で欠陥があると正常にハードウェアが動作しない。この製造不良を発見するものがテストである。製造過程におけるテストは、時間の関係上、クリティカルな入出力値をみるものとなるのが通常である。
- \* ソフトウェアの品質基準は大きく分けてベリフィケーションとバリデーションという二つの考え方がある。ベリフィケーションはソフトウェアプロセスにおいて、各工程が上の工程の品質基準をみたしているかどうかを問題にし、バリデーションはソフトウェアプロセスにおいて生成されるプロダクトが最終的に顧客要求を満たしているかどうかを問題にする。
- \* ソフトウェアのテストにおいては、ソフトウェアプロセスの段階ごとに以下のテスト種類に分けることができる。
  - 単体テスト・結合テスト:前者は関数やモジュール一つごとのテストであり、後者は関数やモジュールを呼び出して行われるテストである。しばしば、スタブやドライバというモックオブジェクトを利用して、テストを行うことが多い。
  - 機能テスト:すべての機能について網羅的に入力値、出力値の組合せが正常かどうかを検証するテスト。
  - システムテスト:システムに負荷をかけたり、動作環境や設定を意図的に変更したり、異常な状況という特殊環境を設定して行われる、品質のロバスト性を検証するテスト。
  - 回帰テスト:ソフトウェアの変更に伴い、コードの先祖返りやデグレードを起こさないように行われるテスト。

### 2) 特許・ライセンス・機密情報などの管理

- \* 競争優位性の保護から機密情報の管理が重要なことは言うまでもない。
- \* 特許を取得しその技術をライセンス提供するビジネスも盛んである。二社間でそれぞれの技術を交換するクロスライセンスもよく行われている。

### 3) 構成管理

- \* ハードウェア・ソフトウェアに関わらず、なんらかのソースコードを複数人で開発する場合や、継続的な上位バージョンの開発には、ソースコードの版管理が重要になる。また、開発に利用するハードウェア要素や構成、ソフトウェア構成などのバージョン、製品仕様等を管理することも必要である。これらを構成管理と言い、ハードウェア要素、ソフトウェア要素の変更に対して、整合性を保つようにコントロールし、常に変更を追跡しなければならない。

スキル区分	OSS モデルカリキュラムの科目	レベル
組み込み分野	24. 組み込みシステムに関する知識Ⅱ	応用
習得ポイント	Ⅱ-24-5. 組み込みシステム開発プロジェクトの体制	
対応する コースウェア	第 11 回 (組み込みコンピュータシステムの開発管理)	

## Ⅱ-24-5. 組み込みシステム開発プロジェクトの体制

組み込みシステム開発プロジェクトにおけるプロジェクトマネジメントに必要な各種のマネジメント、スケジュールマネジメント、リソースマネジメント、要求マネジメント、コミュニケーションマネジメント、について解説を行うと共に、組み込みシステム開発プロジェクトをマネジメントする鍵を提示する。

### 【学習の要点】

- \* 組み込みシステム開発においては、ハードウェアとソフトウェアの摺り合わせを事前に計画し、マネジメントを行う必要がある。
- \* 組み込みシステム開発におけるプロジェクトマネジメントでは、異なったリソースが絡んでいると共に、異なった職能をもつ人材を一つの目的に向かって作業を進めなければならないという難しさがある。
- \* スケジュールマネジメントでは、トレードオフ状況を捌く必要がある。
- \* リソースマネジメントで一番重要な要素は人的資源である。
- \* 要求とプロジェクト活動を統合的にし、すべてのステークホルダーの情報伝達が適切にすることで、プロジェクトが失敗するリスクを減らすことができる。

### 組み込みプロジェクトのポイントとマネジメント

プロジェクトのポイント	必要な要素	必要なマネジメント
ハードウェアとソフトウェアの摺り合わせ	事前の計画	スケジュールマネジメント
	コミュニケーションの円滑化	コミュニケーションマネジメント
	要求の明確化	要求マネジメント
多用なリソースを効率よく活用する	未然にトラブルを防ぐ手法	リスクマネジメント
	トレードオフを捌く手法	リソースマネジメント

図 Ⅱ-24-5. 組み込みシステム開発プロジェクトの体制

## 【解説】

### 1) 組み込みシステム開発におけるプロジェクトマネジメント

- \* 組み込みシステム開発におけるプロジェクトマネジメントとは、ターゲットとする組み込みシステムの開発を計画したコストと時間で達成するために必要なすべての活動を意味する。
- \* 開発プロジェクトのマネジメントで必要なのは次の要素である。
  - プロジェクトの目的とゴールをチームに明示する
  - プロジェクトの計画立案を明示する
  - プロジェクトの状況把握と調整を行う
- \* 組み込みシステム開発においては、ハードウェアとソフトウェアという二つの大きな開発物があるために、これらの摺り合わせを事前に計画し、マネジメントを行わなければ、目標とする機能、性能、品質に到達することが難しくなる。
- \* 組み込みシステム開発におけるプロジェクトマネジメントでは、ハードウェアとソフトウェアという分野にまたがり、異なったリソースが絡んでいると共に、異なった職能をもつ人材を一つの目的に向かって作業を進めなければならないという難しさがある。そのために日々何かしらの問題が生じると仮定したマネジメントが必要になる。

### 2) スケジュールマネジメント

- \* スケジュールマネジメントは定められた期間内で業務を完了するために必要なマネジメントを行うことである。
- \* プロジェクトのスケジュールはプロジェクトの開発コストやリソース、要求事項に密接に関連しているために、リソースが減ったらスケジュールを延ばすなどの、トレードオフを捌かなければならない。
- \* スケジュールの状況を図示するために、しばしばガントチャートやアローダイアグラム等が利用される。

### 3) リソースマネジメント

- \* リソースマネジメントは、人的資源やナレッジ、製造機器、開発ソフトウェアなど、開発に必要な人的、物的リソースをマネジメントすることである。
- \* リソースマネジメントで一番重要な要素は人的資源である。人的資源のマネジメントは、開発チームのメンバーが個々持っている属性である、スキルやナレッジ、そして経験、また稼働可能なスケジュールといったものを、プロジェクトに最適配置することがポイントになる。
- \* 新規採用の場合はその教育や養成をマネジメントすることが必要である。

### 4) 要求マネジメント

- \* 要求マネジメントとは、プロジェクトに対する要求事項とプロジェクトの諸活動の整合をとるために必要なマネジメントである。

### 5) コミュニケーションマネジメント

- \* コミュニケーションマネジメントとは、開発プロジェクトに関係するすべてのステークホルダーの間の情報の伝達が効率よく的確に行うようにマネジメントすることである。

スキル区分	OSS モデルカリキュラムの科目	レベル
組み込み分野	24. 組み込みシステムに関する知識Ⅱ	応用
習得ポイント	Ⅱ-24-6. 組み込みシステムのアーキテクチャ	
対応する コースウェア	第 12 回（組み込みのアーキテクチャ）	

## Ⅱ-24-6. 組み込みシステムのアーキテクチャ

PC/AT 互換機や情報家電、携帯電話、センサシステムなど代表的な組み込みシステムのハードウェアを紹介する。またこれらのシステムにおいて動作するソフトウェア構成の代表例を示し、それぞれの特徴や開発に必要な項目について説明する。

### 【学習の要点】

- \* 代表的な組み込みシステムのハードウェアアーキテクチャは SoC(System on a Chip)である。
- \* 最近の組み込みシステムではメモリアーキテクチャを構成し、多数のメモリの整合性とタイミングを考慮する必要がある。
- \* プロセッサには大きく分けて、汎用プロセッサ、アプリケーション専用プロセッサ、応用指向プロセッサの三つがある。汎用プロセッサでは、OS を載せることが多い。専用プロセッサは高度な処理を行うときに利用される。応用指向プロセッサの代表例は DSP(Digital Signal Processor)や GPU (Graphics Processing Unit)である。
- \* FPGA 等のリコンフィギャラブルコンピューティングを利用したプロセッサが主流になっている。

**プロセッサの種類**

OS	種類/例
汎用プロセッサ	ARM          MIPS PowerPC      SH V850
応用指向プロセッサ	DSP(Digital Signal Processor) GPU (Graphics Processing Unit)
アプリケーション専用プロセッサ	地デジチューナー画像プロセッサ 携帯ベースバンド信号処理プロセッサ

**SoCとしてのFPGAの構造**

論理ブロック  
 入出力ブロック  
 スイッチブロック

アレイ状にならんだ論理ブロックの間にスイッチブロックが配置されている。このスイッチブロックを制御することで、論理ブロック同士の配置を変更できる。

これはロジックを組み替えることと同じである。つまり、結線を自在に組み替えて目的のプロセッサを実現する。

図 Ⅱ-24-6. 組み込みシステムのアーキテクチャ

## 【解説】

### 1) SoC アーキテクチャ

- \* SoC(System on a Chip)アーキテクチャとは、主要な機能を1チップの半導体として実現した組み込みシステムのアーキテクチャである。SoCでは通常のパソコンのマザーボードにある、プロセッサ、バス、メモリ、I/O インタフェース、グラフィックス等を一つの半導体として実現する。
- \* SoC のメリットは以下の点にある。
  - 小さなサイズとなるためにシステムを小さく作ることができる。
  - 物理的に部品点数も少なく、故障が少なくなると共に、保守性が増す。
  - 信号伝達がチップ内だけなので、高性能かつ消費電力を抑えることができる。

### 2) メモリアーキテクチャ

- \* 組み込みでは、リアルタイム処理を実現するために多数のメモリが利用されている。最近では携帯電話や音楽プレイヤーのように大容量のメモリが必要であることから、メモリアーキテクチャを構成し、キャッシュと主記憶、フラッシュメモリ等の整合性とタイミングを考慮する必要がある。
- \* メモリアーキテクチャに依存してソフトウェア構成を考える必要がある。

### 3) プロセッサ

- \* プロセッサには大きく分けて、汎用性の順序で、汎用プロセッサ、応用指向プロセッサ、アプリケーション専用プロセッサの三つがある。汎用性の高さは性能とトレードオフであることが通常である。
- \* 組み込みで仕様される汎用プロセッサは、ARM, MIPS, PowerPC, SH V850 などが主流である。汎用プロセッサは、その上に OS を乗せて利用されることが多く、その場合機能要求をソフトウェアで実現することが多い。そのようなアーキテクチャを採用すると、開発コストが小さくなり、多様な機能開発が実現できる反面、性能が大きく実現できないことがある。
- \* 応用指向プロセッサは汎用と専用の間位置するプロセッサで、ある範囲の幅をもった専用処理を行う。代表的な例では、信号処理に共通な演算処理(例えばフィルタ処理の積和演算など)は DSP(Digital Signal Processor)で行われるのが通常である。また、3D 画像のグラフィックス処理などでは、GPU(Graphics Processing Unit)が利用されるのが通常である。
- \* 専用プロセッサはできる限り高い性能を実現するために、特定のアプリケーションを実現するために利用されるものである。例えば、デジタル放送テレビの放送信号を処理する画像処理エンジンなどは、リアルタイムに高度な性能をもとめられるために、専用プロセッサが利用されている。

### 4) リコンフィギャラブルコンピューティング

- \* SoC においてハードウェアの回路構成を動的に変更できる技術をリコンフィギャラブルコンピューティングと呼ぶ。近年多用される FPGA(Field Programmable Gate Array)が代表例である。
- \* FPGA 等のハードウェアは特定の専用回路ではなく、一種のプログラミングによりハードウェアの機能を実現している。そのために、ソフトウェア開発との相性がよく、ハードウェアとソフトウェアのコーデザインという開発パラダイムという考え方によく合致する。
- \* FPGA 等は高価なマスク処理等が必要ないために、他品種少量生産に向いている。

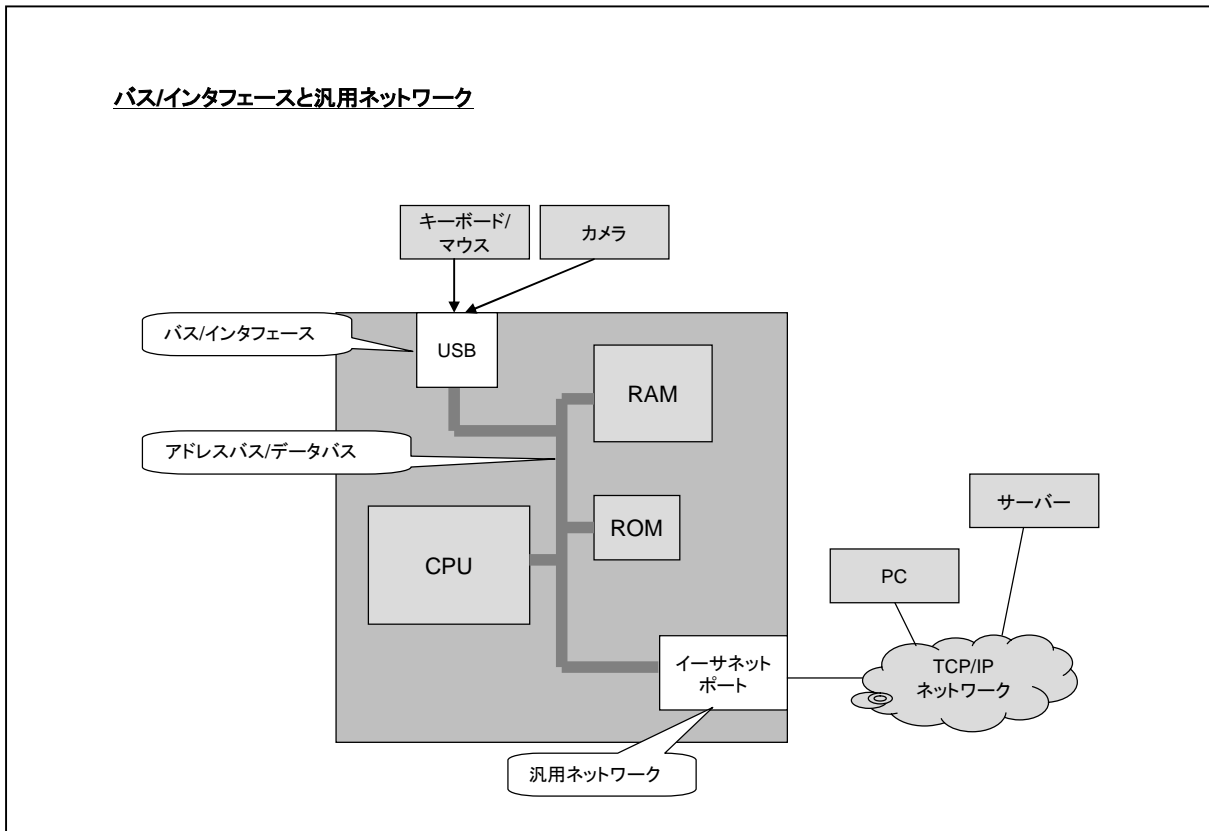
スキル区分	OSS モデルカリキュラムの科目	レベル
組み込み分野	24. 組み込みシステムに関する知識Ⅱ	応用
習得ポイント	Ⅱ-24-7. 組み込みシステムにおけるネットワークのハードウェア要件	
対応する コースウェア	第 13 回（組み込みシステムのネットワーク機能）	

## Ⅱ-24-7. 組み込みシステムにおけるネットワークのハードウェア要件

シリアル通信、赤外線インタフェース、IEEE802.15 などの低速デバイスや、IEEE1394、イーサネット、USB2.0 などの高速デバイスによる通信インタフェースのハードウェアについて、その概要や特徴、利用方法、留意点を解説する。

### 【学習の要点】

- \* 組み込みシステムは、外部ハードウェア機器との接続のために、バス/インタフェースと汎用ネットワークをサポートする必要があることが多い。
- \* バス/インタフェースは CPU に周辺機器を接続するための仕組みであり、主に PCMCIA、PCI、SCSI、USB、GPIB、IEEE1394(FireWire)などが例である。
- \* 汎用ネットワークは、主に CPU をもつ組み込みシステム同士が、相互に通信するためのサービスの仕組みである。イーサネットや IrDA、IEEE 802.11(無線)などが例になる
- \* バス/インタフェースや汎用ネットワークはデバイスドライバを通じて論理的な通信が行われる。
- \* 組み込みシステムにおいては、ハードウェア上の制限から通信容量、電源容量、エラー処理、遅延対策等について十分に検討する必要がある。



図Ⅱ-24-7. 組み込みシステムにおけるネットワーク



## 【解説】

### 1) バス/インタフェースと汎用ネットワーク

- \* 組み込みシステムは、外部ハードウェア機器との接続のために、バス/インタフェースと汎用ネットワークをサポートする必要があることが多い。
- \* バス/インタフェースは CPU に周辺機器を接続するための仕組みであり、ハードウェア同士を低レベルな通信プロトコルにより直接接続する。「ネットワーク」ではない。
- \* 汎用ネットワークは主に CPU をもつ組み込みシステム同士が、相互に通信するためのサービスの仕組みである。イーサネットや IrDA, IEEE 802.11(無線)などが例になる。
- \* 組み込みシステムにおいてバス/インタフェースや汎用通信に対応する場合、しばしばハードウェア上の制限から通信容量、電源容量、エラー処理、遅延対策等について十分に検討する必要がある。

### 2) バス/インタフェース

- \* 組み込みシステムにおける主なバス/インタフェースには次のものがある。
  - PCMCIA:主に IC カードを接続するために利用されるバス。
  - PCI: PC-AT アーキテクチャにおいて現在最も普及しているバス。イーサネットカードやグラフィックカードの接続に利用する。
  - SCSI: ハードディスクなどの様々なハードウェア周辺機器を接続するための汎用バス。
  - USB: 高速シリアルバス規格の一つで、プラグ&プレイおよびホットプラグに対応している。
  - GPIB: 測定機器、計測機器に利用される。エンジニアリングと科学分野で利用が多い。
  - IEEE1394(FireWire): 高速シリアルバス規格の一つで、プラグ&プレイおよびホットプラグに対応している。USB との差はデータ通信にホスト機器が必要ないことである。

### 3) 汎用ネットワーク

- \* 多くの組み込みシステムが汎用のネットワークに接続するようになってきている。そのために汎用の通信インタフェースに対応する必要がでているのが現状である。
- \* 組み込みシステムにおける主な汎用ネットワークには次のものがある。
  - イーサネット: 現在最も普及しているネットワークインタフェース。
  - IrDA: 赤外線を利用したデータ通信を行うネットワークインタフェース。
  - IEEE 802.11: 無線を利用したネットワークインタフェース。イーサネットの無線版。
  - Bluetooth: 近接したデバイス同士の通信を無線で実現する。無線版 USB とも言われる。

### 4) デバイスドライバ

- \* バス/インタフェースや汎用ネットワークはデバイスドライバを通じて論理的な通信が行われる。
- \* デバイスドライバとは入出力装置や外部記憶装置などの周辺デバイスを直接制御するソフトウェアである。
- \* デバイスドライバを利用するメリットは、周辺デバイスの違いを抽象化して、上位にあたるアプリケーションの再利用性を高めることと、アプリケーションの開発自体を容易にするという 2 点である。

スキル区分	OSS モデルカリキュラムの科目	レベル
組み込み分野	24. 組み込みシステムに関する知識Ⅱ	応用
習得ポイント	Ⅱ-24-8. 組み込みシステムにおけるネットワークのソフトウェア	
対応する コースウェア	第 13 回 (組み込みシステムのネットワーク機能)	

## Ⅱ-24-8. 組み込みシステムにおけるネットワークのソフトウェア

組み込みシステムで利用できる各種の通信デバイスに関して、ソフトウェアからの利用方法を説明する。また通信するデータや通信対象、通信手段の選択方法や、組み込みシステムで利用できる通信ミドルウェアについて解説する。

### 【学習の要点】

- \* 組み込みシステムにおけるネットワークのソフトウェアで利用される言語は、C、アセンブラ、C++、Javaの順である。Cとアセンブラの組合せが組み込みソフトウェアの開発において一番利用され、安定している言語セットである。
- \* 組み込みシステムにおけるネットワークのソフトウェアの開発では OS が利用されることが日常化している。
- \* 組み込みインターネットでは通常の汎用 PC と同様のネットワークソフトウェアが利用される。
- \* 組み込みインターネットを利用する場合は OS を利用するが多い。

### 組み込みソフトウェアの使用言語

言語	特徴
C	一番仕様頻度が多い。アセンブラよりパフォーマンスがないが、移植性が非常に高い。
アセンブラ	パフォーマンスがでる。プロセッサごとの開発になり、再利用性が低い
C++	プログラムが肥大化しやすい。
Java	VM(Virtual Machine)のオーバーヘッドが大きく、GC(Garbage Collection)があるために、リアルタイム性を保証できない。

### 組み込みソフトウェアの使用OS

OS	種類
リアルタイムOS	ITRON T-Engine RTLinux VxWorks Windows CE
汎用OS	Linux Windows CE

図 III-24-8. 組み込みシステムで利用される言語と OS

## 【解説】

### 1) 組み込みシステムにおけるネットワークのソフトウェアにおける言語

#### \* C 言語

現在組み込みシステムにおいて最も利用されている言語は C 言語である。C 言語はデバイスドライバなどのハードウェアに直接アクセスするプログラムの作成に向いている。

#### \* アセンブラ

C 言語の次に組み込みで利用されているのがアセンブラである。未だにアセンブラが利用されているのは、割り込み処理などは C 言語を利用していてもアセンブラが必要であること、C 言語よりもパフォーマンスがよいこと、などがあげられる。ただし、命令セットがプロセッサごとに異なるために、ソフトウェアの再利用性が低くなる。

#### \* C++

C++は大規模な組み込みシステムを中心に広がっているが、あまり十分に広がっているとは言えない状況である。その理由として、例外処理やテンプレートなどを不用意に利用してしまうとコードサイズが巨大になってしまうこと、メモリ配置を静的に決められないのでメモリの最適化を意図的に実現することができないこと、などがある。

#### \* Java

Javaは携帯電話のような、マシンが独立で比較的小さなバイトコードをダウンロードして利用する場合に利用されているだけで、十分に広まっていない。その理由は VM (Virtual Machine)のオーバーヘッドが無視できないほど大きいこと、ガベージコレクションによりリアルタイム性を著しく失うことなどがある。

### 2) オペレーティングシステム

\* プロセッサの処理能力性能の向上と、機能要件の肥大化の波が押し寄せた結果、組み込みシステムにおいてオペレーティングシステム(OS)が採用されるケースが急増している。

\* 多くの組み込みシステムではリアルタイム性が求められることから、リアルタイム OS(RTOS)が利用されることが多い。このような RTOS として、ITRON および後継の T-Engine, RTLinux, VxWorks, Embedded Windows CE などがよく使われる。

\* 最近では組み込み機器であっても、プロセッサ性能とメモリ容量が拡大したことから、Linux や Windows の汎用 OS がそのまま搭載されることさえある。

### 3) 組み込みインターネット

\* 携帯電話や情報家電をはじめとして、組み込み機器を利用してインターネット上のサービスを活用する機会が増えてきた。これを組み込みインターネットという。典型的な利用方法が、携帯電話からの Web ブラウジングや動画ストリーミング、メール閲覧である。

\* 組み込みインターネットでは通常の汎用 PC と同様、TCP/IP 通信の上に HTTP プロトコルや IMAP プロトコルを利用したり、UDP/IP の上にストリーミング形式のプロトコルを利用したりする。

\* インターネットに対応するミドルウェアとしてのソフトウェアスタックが必要になるため、その一部として利用可能である OS を採用するケースが多い。

\* 組み込みインターネットに対するソフトウェアスタックが無い場合は、スクラッチから開発することになるが、既にコモディティとなっているインターネットソフトウェアスタックを開発するかどうかは、

十分に検討する必要がある。

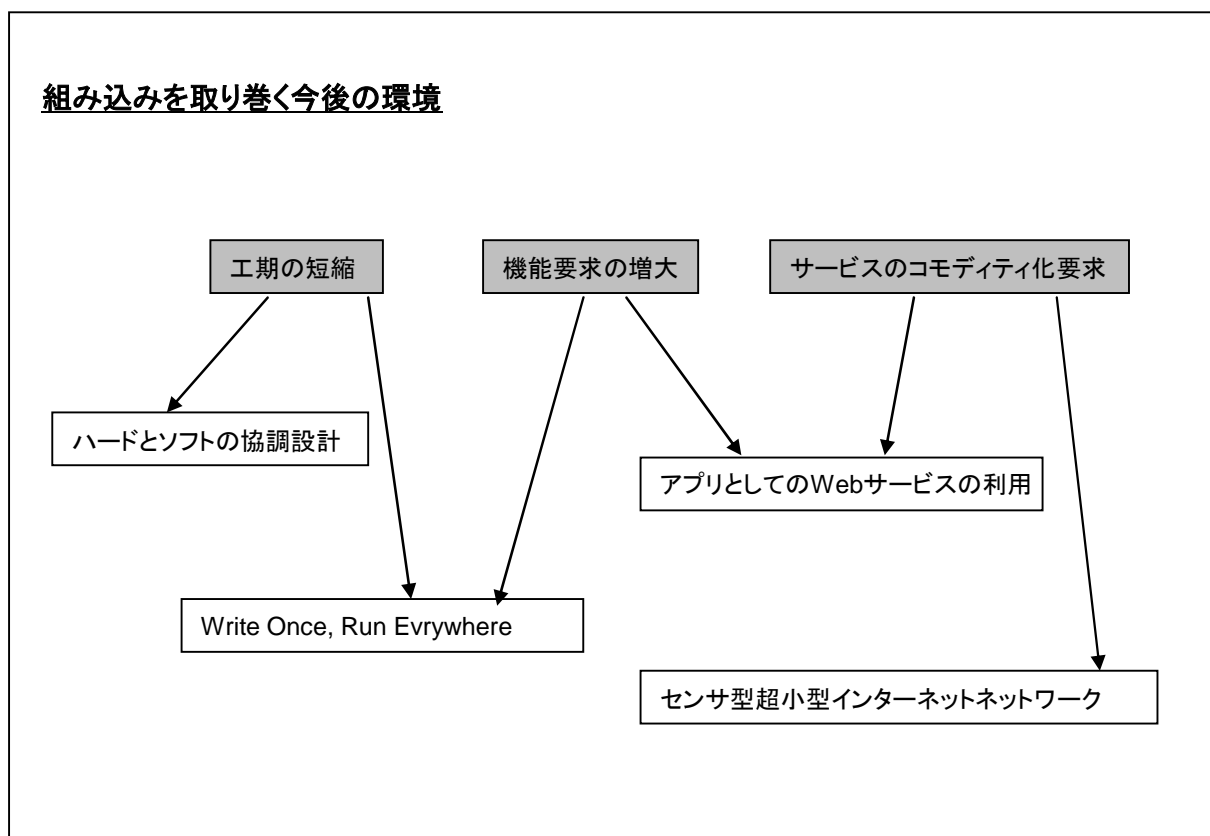
スキル区分	OSS モデルカリキュラムの科目	レベル
組み込み分野	24. 組み込みシステムに関する知識Ⅱ	応用
習得ポイント	Ⅱ-24-9. 組み込みコンピュータを取り巻く今後の環境	
対応する コースウェア	第 15 回 (これからの組み込みシステムの動向)	

## Ⅱ-24-9. 組み込みコンピュータを取り巻く今後の環境

携帯電話、PDA のプラットフォームや情報家電など、今後発展が期待される組み込みシステム応用分野の概要や将来展望、それらの現状と発展の動向など、組み込みシステムを取り巻く様々な環境の変化について解説する。

### 【学習の要点】

- \* 増大する機能要求と工期の短縮を実現するために、IP ベース設計、既存ソフトウェアのライセンス利用、OS の利用はますます強化される。
- \* ハードウェアとソフトウェアの協調設計も強化される。
- \* Write Once, Run Everywhere が何らかの形で実現される。
- \* アプリケーションとしての Web サービスの利用が組み込み端末にも広がる。
- \* センサ型超小型インターネット端末ネットワークにより、新たな情報サービスが生み出される。



図Ⅱ-24-9. 組み込みコンピュータを取り巻く今後の環境

## 【解説】

### 1) 増大する機能要求と工期の短縮

- \* 組み込みコンピュータは、多様化するデバイスインタフェースへの対応、爆発するネットワークサービスへのキャッチアップ、動画等のマルチメディアデータの大量処理というように、ますますの多機能化とデータ処理の速度の増加を求められている。
- \* その反面、サイズの縮小化や消費電力の低下、そしてマーケットドリブンな工期の短縮化といったことを求められている。
- \* 開発工数が大きくなるということと、工期を短縮しなければならないという矛盾した状態を解消するために、今後も以下の既存モジュールの利用を軸とした組み込み開発は増えていくだろう。
  - IP ベース設計
  - 既存ソフトウェアのライセンス利用
  - OS の利用

### 2) ハードウェアとソフトウェアの協調設計

- \* 現在においても SoC の設計に利用される言語は C 言語であり、ソフトウェアアプリケーションの実装に利用される言語も C 言語である。このように、将来はハードウェアとソフトウェアの境界を明確に定めることがない、より緊密な協調設計が可能になり、マーケットの要求に素早く応えることが期待できる。

### 3) Write Once, Run Everywhere の実現

- \* Write Once, Run Everywhere とは一度プログラムを書くだけで、どんなシステムでもそのプログラムが動作するという考え方である。現在 Java 言語はある程度これを実現してはいるが、それでもヴァーチャルマシンのプロセッサ依存やその性能保証という点で十分に機能しているわけではない。
- \* 今後、ますますプロセッサ性能とメモリが増大し、かつ既存ソフトウェアや汎用 OS の利用が広がれば、真の Write Once, Run Everywhere が実現できると期待できる。

### 4) アプリケーションとしての Web サービスの利用

- \* インターネットサービスは今後もますます充実すると考えられる。このために、インターネット用のブラウザアプリケーションがあるだけで、多種多様なサービスを自動的に享受できることになっていく。
- \* これは、組み込み機器独自のアプリケーション開発をする必要がないということになってくるために、コストの著しい削減と効用の増大につながる。従って、組み込み機器のインターネット対応の傾向はますます強化されるだろう。

### 5) センサ型超小型インターネット端末ネットワーク

- \* 今後様々な社会環境や自然環境の状況を刻々と報告する、センサ型の超小型インターネット端末が組み込みシステムとして登場することは容易に考えられる。
- \* このセンサ型超小型インターネット端末が巨大なセンサネットワークとなり、そのネットワークを利用してより細かくパーソナルな情報サービスを提供することになるだろう。

スキル区分	OSS モデルカリキュラムの科目	レベル
組み込み分野	24. 組み込みシステムに関する知識Ⅱ	応用
習得ポイント	Ⅱ-24-10. インターネット以外の組み込みコンピュータネットワーク	
対応する コースウェア	第 15 回 (これからの組み込みシステムの動向)	

## Ⅱ-24-10. インターネット以外の組み込みコンピュータネットワーク

携帯電話による通信や、赤外線通信、ユビキタスネットワークなど、組み込みシステムで利用できるインターネット以外のネットワークアーキテクチャについて解説する。さらに組み込みシステムにおける IPv6 の利用動向についても述べる。

### 【学習の要点】

- \* 携帯電話による通信や赤外線、RFID によるユビキタスネットワークなど、インターネットを離れてもネットワークは広く利用されている。
- \* 組み込みの分野で市場規模の大きいネットワークの例としては、自動車の車載ネットワークと工場利用される FA システムがあげられる。
- \* IPv6 の普及によりあらゆる機器に IP アドレスを持たせインターネットによる通信が可能となることから、現在は通信機能のない家庭用電化製品への適用など、幅広い利用が期待されている。

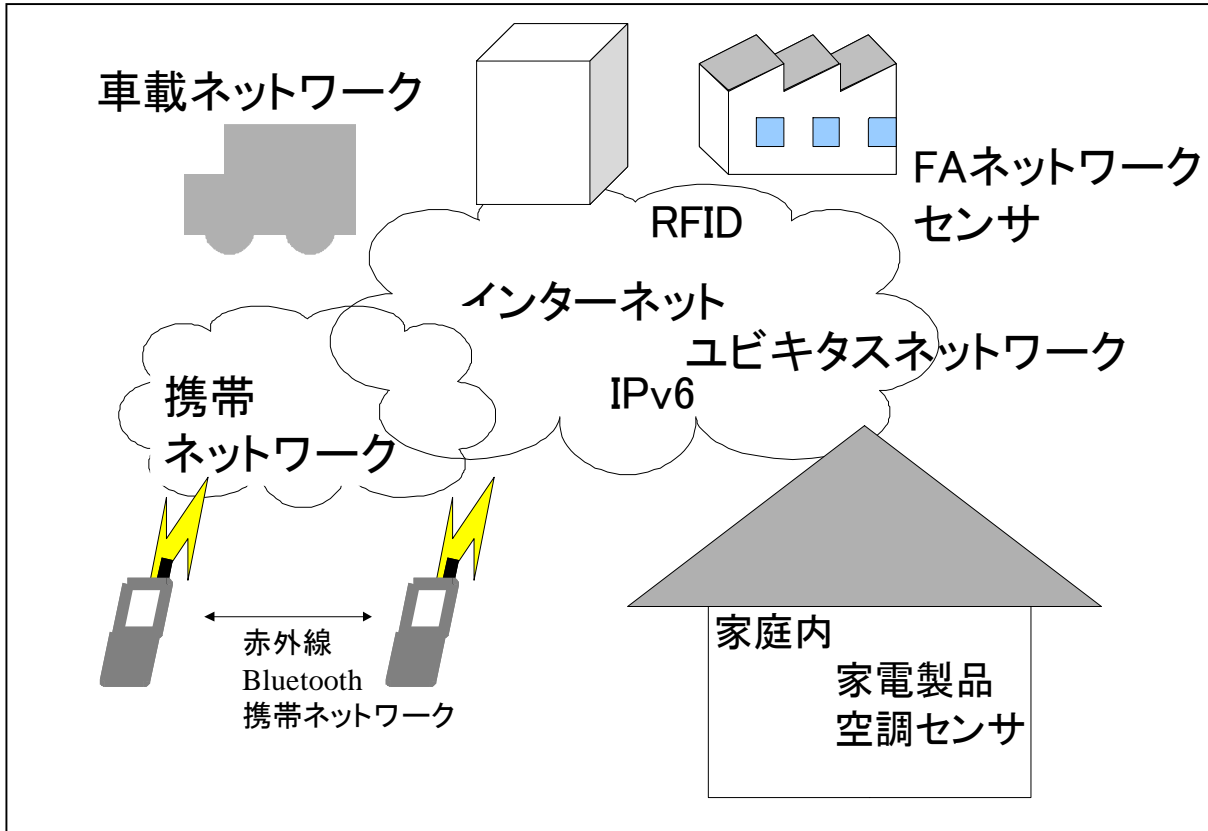


図 Ⅱ-24-10. 広がる組み込みコンピュータのネットワーク

## 【解説】

### 1) インターネット以外の組み込みコンピュータネットワークと IPv6

- \* 組み込みシステムにおいても、他のシステムと同様に、赤外線通信、ユビキタスネットワークなどのインターネット以外のコンピュータネットワークが利用されている。
- \* IPv6 の普及により、全ての機器やデバイスが個別に IP アドレスを持ちインターネットを利用した通信の実現が可能となり、期待されている。
- \* 組み込みシステムにおける IPv6 の利用は、TCP/IP プロトコルスタックの実装をハードウェア (ROM 化)で行う場合と、ソフトウェアで行う場合がある。ソフトウェアで行う場合は、OS を利用する場合が大多数である。
- \* インターネット以外の組み込みコンピュータネットワークで市場シェアとして大きいものは、自動車や車両に用いられる車載ネットワークと、生産工場で利用される FA ネットワークである。

### 2) 車載ネットワーク

- \* 組み込みシステムとして自動車を見た場合、組み込みインターネットなどの情報ネットワークに比べて、高度の信頼性、リアルタイムの保証性が必要とされる。この車載ネットワークについては国際標準化が行われ、オープン化が進行している。
- \* 自動車向けの制御ネットワークとして国際標準として利用されているのが CAN(Controller Area Network)である。
- \* アクチュエーターやセンサなどのデバイスをつなぐための低速・低コストのネットワークとしては、LIN(Local Interconnect Network)が利用されている。
- \* CANはマルチマスタ方式のバス型ネットワークである。一つのバスに複数のノードが繋がれており、通信はすべてのノードにメッセージのブロードキャストが行われる。メッセージには ID がついており、この ID が優先度を表す。各ノードは ID をみて自分に必要なメッセージを取り込む。
- \* CAN においては、ブロードキャストと優先度の確保がリアルタイムの保証を担保することになる。

### 3) FA ネットワーク

- \* FA システムのネットワークも高度の信頼性とリアルタイム保証が求められる分野である。
- \* FA システムの組み込みシステムネットワークについては、次の三つのレベルに分けて考えることができる。
  - コンピュータレベル:工場全体の管理を行う。
  - コントローラレベル:生産ラインごとの管理・制御。
  - デバイスレベル:一つの製造機器における制御コンピュータとセンサ・アクチュエータの管理。
- \* コンピュータレベルのネットワークには Ethernet を利用して TCP/IP 経由の情報系ネットワークが利用されることが多い。
- \* DeviceNet(<http://securities.jp/ODVA/devicenet/>)はコントローラ及びデバイスレベルのネットワークで、車載ネットワークである CAN をそのまま流用する。
- \* CC-Link はコントローラ及びデバイスレベルのネットワークで、マスタスレーブ型のバス型ネットワークである。CC-Link は日本の業界団体による普及活動が行われている。