

第3章 モデルカリキュラム

会員企業アンケートに基づき、C言語による組込みプログラム開発の基本技術を習得することを目的として Off-JT ならびに OJT のモデルカリキュラムを平成 20 年度に作成し、平成 21 年度 JASA 会員企業 28 社 314 名が実践型人材養成システムを活用した。

平成 21 年度下期、本訓練を導入した 12 社に対してカリキュラムに関して再度アンケートを実施し、モデルカリキュラムの改訂を行った。

1. モデルカリキュラムの開発

座学等（Off-JT）及び実習等（OJT）の訓練科目の開発にあたっては、ETSS と ETEC を利用して、実践型人材養成システムの評価の方法として利用する評価シート（ジョブ・カード様式 6）の項目も有機的にかつ並行的に開発した。

1) スキル標準（ETSS）の活用

独立行政法人 情報処理推進機構（IPA）が策定する組込みスキル標準（ETSS）のキャリアフレームワークを元に対象職種の選定とモデルカリキュラムの項目設定を行った。

※ 組込みスキル標準（ETSS）とは

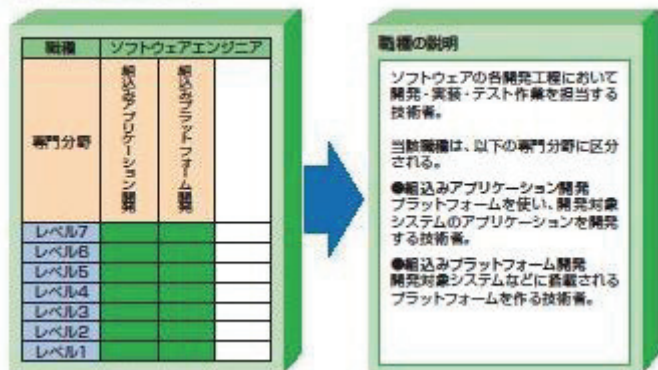
独立行政法人 情報処理推進機構（IPA）によって定められた組込み技術者のスキル指標のことで、10 職種・7 段階（レベル）に分類し、必要なスキル領域を定めている。（右図参照）

ETSS は、どの職種のどのレベルのスキルを有しているかを客観的に判断するための指標であり、能力スキルアップの指標として、研修プログラム策定の指標として利用されている。

職種	プログラマー	プロジェクトマネージャ	システムエンジニア	システムアーキテクト	ソフトウェアエンジニア	プリッジSE	開発エンジニア	開発エンジニア	QAエンジニア	テストエンジニア
専門分野	組込みシステム開発	組込みソフトウェア開発	組込み開発技術	組込みアプリケーション開発	組込みプラットフォーム開発	組込みアプリケーション開発	組込みプラットフォーム開発	組込みソフトウェア開発	組込みソフトウェア開発	組込みシステム開発
レベル 7										
レベル 6										
レベル 5										
レベル 4										
レベル 3										
レベル 2										
レベル 1										

ソフトウェアエンジニアの概要（右図を参照）を元に、モデルカリキュラムが目指す組込みソフトウェア技術者の職種はソフトウェアエンジニアに分類されると特定した。

ソフトウェアエンジニア



ソフトウェアエンジニアのスキル領域（下図を参照）を参考に、教育訓練目標を下記の通り定めた。

- (ア) 組込みソフトウェアの基本技術を習得する。
- (イ) ソフトウェア詳細設計方法を習得する。
- (ウ) ソフトウェアコード作成方法を習得する。
- (エ) プログラムテスト方法を習得する。

専門分野	スキル領域	
	職種共通スキル項目	専門分野固有スキル項目
組込みアプリケーション開発	●開発技術 システム要求分析、システム方式設計、ソフトウェア要求分析、ソフトウェア方式設計、ソフトウェア詳細設計、ソフトウェアコード作成とテスト、ソフトウェア結合、ソフトウェア適合性確認テスト、システム結合、システム適合性確認テスト ●管理技術 統合マネジメント、スコープマネジメント、タイムマネジメント、コストマネジメント、品質マネジメント、組織マネジメント、コミュニケーションマネジメント、リスクマネジメント、調達マネジメント、開発プロセス設定、知識マネジメント、開発環境マネジメント、構成管理・変更管理 ○パーソナルスキル コミュニケーション、リーダーシップ、ネゴシエーション、問題解決	●技術要素 プラットフォーム、開発で必要となる技術要素
組込みプラットフォーム開発		●技術要素 プラットフォーム（作るスキル）、開発で必要となる技術要素

注：スキル領域におけるスキル項目は、職種におけるモデルレベルにて定められるスキル項目を記載

また、座学等及び実習等で習得する技能並びにこれに関する知識または職務の範囲を下記の通り定めた。

- ① 組込みソフトウェア開発技術者として、上長の指導の下にプログラミング、ソフトウェア単体テストが遂行するために必要な技能、技術及び知識
- ② 組込みスキル標準（ETSS）のスキル項目について、レベル1相当のスキル

ETSS に関して詳しい情報は、独立行政法人情報処理推進機構のホームページを参照

<http://sec.ipa.go.jp/ETSS/index.html>

2) 組込みソフトウェア技術者試験(ETEC)の活用

組込み技術知識レベルを評価する、JASA が主催する組込みソフトウェア技術者試験 (ETEC) を座学等 (Off-JT) の評価基準として、活用することとした。

以下がその評価要素である。

技術要素

第1階層	第2階層	第3階層	スキル項目
プラットフォーム	プロセッサ	プロセッサコア	MPU、バス、レジスタセット、RISC、CISC、DSP、GPU、MMU、省電力制御、パイプライン、スーパスカラ、など
		プロセッサ周辺メモリ	割込み、タイマ/カウンタ、DMA、WDT、キャッシュ、など
	基本ソフトウェア	ブート	ROM、RAM、Flashメモリ、メモリインタリーブ、デュアルポートメモリ、など
		カーネル	ROM化、ブートルーディング、スタートアップルーチン、タスク、共有ルーチン、システムコールサービス、同期、排他制御、デッドロック、など
	支援機能	デバッグ機能	ICE、JTAG、ソフトデバッグ、オシロスコープ、ロジアナ、ログ収集/解析、など

開発技術

第1階層	第2階層	第3階層	スキル項目
ソフトウェア詳細設計	ソフトウェアの詳細設計	設計手法	分割、モジュール化、隠蔽化、フローチャート、タイミングチャート、UML、状態遷移図、設計ツール、など
		信頼性設計、実時間設計	QoS、誤り検出、など
	ソフトウェア詳細設計のレビュー	レビュー	レビュー手法、など
ソフトウェアコード作成とテスト	プログラム作成とプログラムテスト項目の抽出	プログラミング	C言語に関すること、コーディング規約、MISRA-C、プログラミング技術、チューニング技術、クロス開発、オブジェクトモジュール、静的解析ツール、カバレッジ、同値分割、など
	コードレビューとプログラムテスト項目のデザインレビュー	レビュー	レビュー手法、など
	プログラムテストの実施	プログラムテスト	カバレッジ、同値分割、ホワイトボックステスト、ドライバ、スタブ、自動化テスト、テストツールなど
ソフトウェア結合	ソフトウェア結合テスト仕様の設計と実施	-	テスト環境設計/構築、テストツールの選定、直交表、カバレッジ、自動化テスト、など
クロス開発技術	-	-	オブジェクトファイルフォーマット、PIC、リロケータブルファイル、コードチューニングなど

管理技術

第1階層	第2階層	第3階層	スキル項目
プロジェクトマネジメント	品質マネジメント	計測	品質特性、ソフトウェアメトリクス、など
開発プロセスマネジメント	構成管理・変更管理	目的	構成管理の目的、など

3) 座学等(Off-JT)の研修項目の確定

1)-1, 1)-2 を元に、座学等(Off-JT)訓練科目として下記項目を選定した。

<学科>

コンピュータ基礎
ソフトウェア開発基礎
C言語基礎
組込みシステム基礎
リアルタイムOS
ソフトウェア設計手法

<実技>

C言語実践
組込みプログラミング
プロジェクト型演習

4) 座学等(Off-JT)の時間数確定

① 時間数の算定の基準 1

選定した座学等(Off-JT)訓練科目について JASA の会員企業に実施したアンケートを時間数算定の参考とした。

Q3-3 新入社員研修プログラムの希望日数		
	総日数	平均 36 日
コンピュータ基礎		平均 3 日
ソフトウェア開発基礎		平均 3 日
C 言語基礎		平均 3 日
C 言語実践		平均 6 日
組込みシステム基礎		平均 3 日
組込みプログラミング		平均 5 日
リアルタイム OS		平均 4 日
状態遷移設計		平均 2 日
プロジェクト演習		平均 7 日 /有効回答数 16 件

*一日の訓練時間は 8 時間を想定

② 時間数の算定の基準 2

アンケート結果と予定した指導内容を考慮し、モデルカリキュラムを策定した。

組込み技術者養成コース	総日数
	40 日
コンピュータ基礎	2 日
ソフトウェア開発基礎	3 日
C 言語基礎	4 日
C 言語実践	7 日
組込みシステム基礎	2 日
組込みプログラミング	7 日
リアルタイム OS	4 日
ソフトウェア設計手法	2 日
プロジェクト演習	9 日

*一日の訓練時間は 8 時間を想定

【各訓練科目の時間数算定に関する補足説明】

コンピュータ基礎・・・対象とする範囲が非常に広いため、ワーキンググループでは、組込み開発に必要となる訓練項目を重点的に 2 日間のカリキュラムとして策定した。

C 言語基礎・・・・・・アンケート対象企業は理系卒採用が多かったが、文系卒受講者も訓練対象となることを考慮し、アンケート結果に 1 日増しの 4 日間とした。

C 言語実践・・・・・・C 言語基礎同様に、アンケート対象企業は理系卒採用が多かったが、文系卒受講者も訓練対象となることを考慮し、アンケート結果に 1 日増しの 7 日間とした。

組込みシステム基礎・・・・・・コンピュータ基礎との訓練項目の重複を考慮し、また各講座で実践的に取り扱う理由項目を検討した結果、2 日間とした。

組込みプログラミング・・・・アンケート結果では 5 日程度であったが、Off-JT 後における OJT が効率的に実施できるように、本訓練では基礎的な知識を反復して習得することをねらいとして 7 日間の十分な時間を設定した。

ソフトウェア設計手法・・「状態遷移設計」だけでなく、もっと幅広く設計手法を習得させるべく、コース名も「ソフトウェア設計手法」に修正。日数は原案通りとした。

プロジェクト演習・・この演習の重要性は、ETSS 概説書の教育研修基準の中でも述べられている通りであり、アンケート結果に 2 日増しの 9 日間とした。

5) 実習等(OJT)の研修項目の確定

1)-1, 1)-2 を元に、実習等(OJT)の訓練科目として下記項目を選定した。

ソフトウェア詳細設計

ソフトウェアコード作成

プログラムテスト

※実習等（OJT）の訓練時間数について

実習等（OJT）の時間数については、上記で定めた実習項目については各企業、各配属先で主たる業務が異なるため、業務の実情に合わせた時間配分設定を申請企業ができるよう、JASA では確定していない。

2. モデルカリキュラムの改定点

平成 21 年度本訓練実施企業のアンケートを元に、次の通り、改定を実施した。

コース名	組込み技術者養成コース
到達目標	上長支援の下に、チームでの開発作業を遂行できるレベル (ETSSスキルレベル1 = ETEC クラス2グレードB (2B) 取得できるレベル)
対象者	日常のパソコン利用において、キーボード操作、メール送受信、Web検索ができること

■ 基礎座学科目 6科目 推奨訓練時間計136時間

科目	概要	具体的な項目例	時間
コンピュータ基礎	コンピュータで扱う数値と単位の理解 コンピュータ機器の種類と特徴の理解 コンピュータで扱う計算方法の習得 論理的な演算や回路に関する基本概念の習得 組込みシステム概要の理解	基数と基数変換/2進法/16進法/加減算と負数表現/ 論理シフトと算術シフト/小数の表現/浮動小数点と誤差/ 情報の単位/文字表現/コンピュータの機器構成/ コンピュータの種類と特徴/CPUアーキテクチャ/メモリアーキテクチャ/補助記憶装置/入力アーキテクチャと周辺装置/論理演算/論理回路/エンベデッドシステム	16時間
	↓ 概要の一部削除 ↓	↓ 項目例の一部削除 ↓	
	コンピュータで扱う数値と単位 コンピュータ機器の種類と特徴 コンピュータで扱う計算方法 論理的な演算や回路に関する基本概念	基数と基数変換/2進法/16進法/加減算と負数表現/論理シフトと算術シフト/小数の表現/浮動小数点と誤差/情報の単位/ 文字表現/コンピュータの機器構成/コンピュータの種類と特徴/ CPUアーキテクチャ/メモリアーキテクチャ/補助記憶装置	
改定ポイント	アンケートにおいても「必要かつ十分な時間と内容である」と評価されていたので、特に大きな内容変更は行わず、文言をわかりやすく修正する程度にとどめた。		
ソフトウェア開発基礎	ソフトウェア開発の基本的知識の習得 開発工程の理解 オブジェクト指向の理解 データベースに関する基礎知識の習得 ネットワーク基礎知識の習得	ソフトウェアとOS/マルチプログラミング/実記憶管理/ 仮想記憶システム/プログラム言語/コンパイラ/開発モデル/ コスト見積と工程管理/外部設計・内部設計/プログラム設計とプログラミング/テストの種類と技法/オブジェクト指向/データベースとは /DBMS/リレーショナルデータベース/正規化/ER図/SQL/正規化演習/ ER図/SQL演習/QSI参照モデル/プロトコル/MACアドレス/IPアドレス/サブネット/DHCP/デフォルトゲートウェイ	24時間
	↓ 科目名の変更 ↓	↓ 項目例の一部削除と「フローチャート」の追加 ↓	
	プログラム開発基礎	ソフトウェア開発の基本的知識 開発工程の基礎	
改定ポイント	①科目名の変更：「ソフトウェア開発基礎」という訓練科目名は、網羅する範囲が広すぎるという理由から、「プログラム開発基礎」と変更し、組込み開発技術者養成という目的を明確にした。 ②削除項目：アンケートでデータベースに関連する項目に対して不要という意見が多く寄せられたので、データベースに関する項目を削除した。		
C言語基礎	アルゴリズムや基本文法を習得 ポインタ・構造体の基本的な理解 ビット演算・シフト演算の習得 自作関数の作成プロセスの習得 配列・文字列の扱い方・変数の理解 条件分岐の習得	変数/基本制御構文/ポインタ/構造体/ 自作関数/定数/マクロ定義/配列/文字列/型変換/ メモリ動的確保/ビットシフト/プロトタイプ宣言/ファイル分割/ 関数名ポインタ/引数のポインタ渡し/メンバ構造体/アルゴリズム基礎/リスト/スタック/キュー/線形探索/二分探索/ソート	32時間
	↓ 概要の表現変更 ↓	↓ 項目例の一部削除と以下の追加 ↓ 「記憶クラス」「標準ライブラリ」	
	アルゴリズムや基本文法 ポインタ・構造体 ビット演算・シフト演算 自作関数の作成プロセス 配列・文字列・変数 条件分岐	変数/記憶クラス/基本制御構文/ポインタ/構造体/自作関数/ 定数/マクロ定義/配列/文字列/型変換/メモリ動的確保/ ビットシフト/プロトタイプ宣言/標準ライブラリ関数/ アルゴリズム基礎/リスト/スタック/キュー	
改定ポイント	アンケートでは「C言語基礎はモデルカリキュラムの中でも重要な項目。プログラミングの基礎技術をしっかりと習得させることが必要」との意見が多かった。この結果を踏まえ、「記憶クラス」「標準ライブラリ」など基礎として必要な項目を追加するとともに、あまり実践では使わない一部の項目を削除し、全体として基礎教育の充実を図った。		
組込みシステム基礎	組込みシステムの特徴 組込みシステムの開発方法 開発ツール（マイコン）の利用方法	組込みシステムの特徴/組込みシステムの開発方法/ マイコンアーキテクチャ/割込み処理/マイコンの入出力	16時間
		↓ 以下の項目例の追加 ↓ 「メモリ」「タイマ」「リセット」「論理演算」「論理回路」	
		組込みシステムの特徴/組込みシステムの開発方法/ マイコンアーキテクチャ/割込み処理/マイコンの入出力/ メモリ/タイマ/リセット/論理演算/論理回路	
改定ポイント	追加項目：アンケートより「IT系の発想を組込み系の発想に転換させるため、マイコンのアーキテクチャを理解させる必要がある」との意見があった。そのため、マイコンの基本的な動きの理解を深めるため、「メモリ」「タイマ」「リセット」等を追加した。		

科目	概要	具体的な項目例	時間
リアルタイムOS	リアルタイムOS概要 リアルタイムOSの利用方法 システムコール リアルタイムOSの下でのドライバの作成方法	リアルタイムOSの概要/マルチタスク/リアルタイム/タスク/ タスク管理関連システムコールの実習/イベントフラグ/ セマフォ関連システムコールの実習/ データキューを使ったタスク間通信の実習/ デバイスドライバの作成実習	32時間
↓ 科目名の変更 ↓	↓ 概要の表現変更 ↓	↓ 項目例の一部表現の変更と以下の追加 ↓ 「メールボックス」「タイマ」「メモリプール」 「マルチタスクデバッグとソフトウェア検証」	↓ 時間減
リアルタイムOS基礎		リアルタイムOSの概要/マルチタスク/リアルタイム/タスク/ タスク管理関連システムコール/イベントフラグ/ セマフォ関連システムコール/データキューを使ったタスク間通信/ メールボックス/タイマ/メモリプール/ マルチタスクデバッグとソフトウェア検証/デバイスドライバ	24時間
改定ポイント ①科目名の変更:「リアルタイムOS」は基礎座学科科目に分類されるため「リアルタイムOS基礎」と変更した。 ②時間の削減:アンケートより、プログラム開発基礎においてデータベースに関連する項目が不要という意見が多く寄せられた結果と同様に、リアルタイムOSも必要・不必要が分かれる項目であった。よって、従来の時間数より8時間削減とした。 ③追加項目:アンケートより「マルチタスク・マルチスレッドは初心者には分かりづらい部分であるため、タスクの切り替えなどは概念での説明だけではなく、実際のシステムを用いて割り込みフラグ、メモリの内容、レジスタ値の遷移を1ステップずつ確認させるべき」とあった。よって、「マルチタスクデバッグとソフトウェア検証」等の項目を追加した。			

ソフトウェア設計手法	代表的な設計手法の理解	状態遷移図・状態遷移表による設計/モジュール分割手法/ オブジェクト指向の開発手法 ↓ 項目例の表基準の変更と、「UML」の追加 ↓ モジュール分割手法/オブジェクト指向の開発手法/UML/ 状態遷移図・状態遷移表による設計	16時間 ↓ 時間増 24時間
改定ポイント ①時間の拡張:アンケートの「UMLをプラス1日に入れた方が良い」の多数意見より、従来の時間数より8時間の拡張とした。 ②追加項目:アンケートより「オブジェクト指向開発手法では「UML」をぜひとも追加してほしい」という意見が非常に多く「UML」を追加した。			

■ 演習科目 3科目 推奨訓練時間計184時間

科目	概要	具体的な項目例	時間
C言語実践	入力/表示を体得。 構造体の利用と関数化手法を習得。 ファイルの取り扱い	キーボード入力と合計加算/棒グラフ表示/構造体のソートと表示/ ファイルの読み込みと単語数カウント/ チェーン構造によるキュー管理	56時間
↓ 科目名の変更 ↓	↓ 概要の表現変更 ↓	↓ 項目例の追加 ↓ 「高度な演算子の利用」「ビットフィールド」「自己参照構造体」 「関数ポインタ」「引数のポインタ渡し」「ファイルIO」 「分割コンパイル」	
C言語プログラミング	入力/表示 構造体の利用 関数化手法 ファイルの取り扱い	高度な演算子の利用/ビットフィールド/関数ポインタ/ 引数のポインタ渡し/自己参照構造体/ファイルIO/ 分割コンパイル/キーボード入力と合計加算/棒グラフ表示/ 構造体のソートと表示/ファイルの読み込みと単語数カウント/チェーン 構造によるキュー管理	
改定ポイント アンケートでは、科目名の「実践」という表現と講義項目のギャップを指摘する声があった。そこで「C言語基礎」では取り扱 わない、より高度な言語仕様を導入したうえで、基礎的なアルゴリズムを用いたプログラミング演習を通してC言語による表現 を身につける演習科目として科目名を「C言語プログラミング」と改め、項目例についても「C言語基礎」からの移動と若干の 追加を行った。			

組み込みプログラミング	アセンブリ言語 C言語を使ったマイコン内蔵IO利用方法の理 解 C言語を使った周辺IO利用方法の理解	アセンブリ言語の概要 アセンブリ言語を使ったプログラム実習 組み込みで使うC言語の特徴 マイコン実習(SWスキャンによるLED点滅/SW割込みによるLED点滅/ タイマによるLED点滅/LCD表示/PWM制御によるモータ回転数制御)	56時間
	↓ 概要の表現変更 ↓	↓ 項目例のうち「マイコン実習」の内容例の記述変更 ↓	
	アセンブリ言語 C言語を使ったマイコン内蔵IO利用方法 C言語を使った周辺IO利用方法	アセンブリ言語の概要/アセンブリ言語を使ったプログラム実習/ 組み込みで使うC言語の特徴/マイコン実習(スイッチ入力/LED点滅/割 込み/タイマ/LCD表示/PWM制御)	
改定ポイント マイコンを用いて実際にI/O制御を行う実習を中心とした演習科目として、科目概要および項目例の表現を整理した。			

プロジェクト型演習	V字開発工程 開発プロセスの疑似実習	要求仕様の提示 要求仕様分析/構造化設計&レビュー 詳細設計&レビュー プログラミング 単体テスト/システムテスト プレゼンテーション(発表会)	72時間
		↓ 項目例の一部に「レビュー」を追加 ↓	
		要求仕様の提示 要求仕様分析/構造化設計&レビュー 詳細設計&レビュー プログラミング&レビュー 単体テスト/システムテスト&レビュー プレゼンテーション(発表会)	
改定ポイント アンケートの要望を取り入れてすべての工程ごとにレビューを設けることで、より現実に近いプロジェクトの流れを経験できる 擬似プロジェクトとして内容の充実を図った。			

<参考>推奨されるOJT科目と内容項目例

科目	概要	具体的な項目例	時間
科目	具体的な内容項目例		
ソフトウェア詳細設計	設計基本技法/構造化設計/設計ツール/オブジェクト指向設計/OS仕様/デバイス仕様/モジュール処理内容/データ構造詳細化/ユニット構成図/状態遷移による表現/CASEツールの使用/詳細設計書の作成 ↓項目例の一部(4項目)を削除し、より具体的なOJT項目例を盛り込んだ	任意	
	デバイス情報の収集・適正判断/非機能要件を満足する詳細設計/モジュール構造化/OS仕様への対応/デバイス仕様への対応/モジュール処理内容の設計/データ構造詳細化/ユニット構成図/状態遷移による表現/CASEツールの使用/プログラムテスト方針策定/詳細設計書の記述作成/詳細設計書のレビュー/レビュー結果のまとめ/レビュー有効性のアセスメント		
改定ポイント	アンケートでは特に大きな指摘事項はなかった。評価シートとの整合性を確保するため、Off-JTで学ぶ知識項目を削除し、評価シートを参考とし、開発業務で必要となるスキル項目を追加した。		

ソフトウェアコード作成	プログラミング技法/言語仕様/コーディング規約/構築ツールの使用/静的解析ツールの使用/ソフトウェアコードの生成 ↓項目例の一部(1項目)を削除し、より具体的なOJT項目例を盛り込んだ。	任意
	ソフトウェア開発環境の構築と使用/コーディング規約の適用/ソフトウェアコードの作成/コード静的解析ツールの使用/ソフトウェアコードのレビュー/構成管理ツールの使用	
改定ポイント	上記と同様に、評価シートの項目との整合性を確保するため、開発業務で必要となる項目を追加した。	

プログラムテスト	ホワイトボックステスト手法/テスト環境の構築と使用/テスト項目の抽出/不具合情報の収集/不具合の特定/コード修正 ↓項目(「ブラックボックステスト」)を追加し、より具体的なOJT項目例を盛り込んだ。	任意
	ソフトウェアコードのレビュー/ホワイトボックステスト/ブラックボックステスト/テスト環境の構築と使用/テスト項目の抽出と妥当性の確認/テストデータの作成/テストツール・デバッグツールの選定・構築・使用/テスト実施/不具合情報の収集/不具合の特定/コード修正/テスト結果報告書の作成	
改定ポイント	アンケートではブラックボックステストを追加する指摘があり、追加した。上記と同様に、評価シートの項目との整合性を確保するため、開発業務で必要となる項目を追加した。	

3. モデルカリキュラム (Off-JT) の詳細

平成 21 年に本訓練を導入した企業のコメントを元に、前述「2.モデルカリキュラムの見直し」を介して、以下の通りモデルカリキュラム(Off-JT)詳細が出来上がった。

本カリキュラムは、パソコンを触ったことがあるレベルから、プログラムに必要な基礎知識、開発言語(C 言語)の学習・演習、組込み開発の学習・演習を積み重ね、最終段階で「プロジェクト型演習」のチーム開発の体験を通して、学んだ知識をアウトプットすることで知識の定着化と未習熟部分を認識し、また開発プロセスの全容とチームワークを体得することで、現場での OJT に向けた準備を行う。

コンピュータ基礎

狙い： IT 技術者に共通する基本的な知識の習得を目指す。

訓練目標： コンピュータ基礎の理解

前提知識： 一般的な PC 操作ができること

訓練詳細(例)：

- 情報システムに携わる仕事(業務システムエンジニアと組込み技術者)
- 組込みシステム概要
- データ表現(2 進数、16 進数、加減算と負数表現、基数変換)
- 文字表現
- 論理演算・論理演算回路
- コンピュータ基本構成
- 入出力装置
- 主記憶装置
- 補助記憶装置
- CPU の動き
- ソフトウェアの種類
- OS の役割
- プロセス管理
- メモリ管理
- 入出力管理
- ファイル管理
- DB 概要
- ネットワーク概要

推奨訓練時間： 16 時間

プログラム開発基礎

狙い： ソフトウェア開発で必要となる基本的な知識の習得を目指す。

訓練目標： 1. ソフトウェア開発(プログラム開発)の基本的な知識の習得
2. 開発工程の基礎知識の習得

前提知識： 基本的なコンピュータ知識があること。一般的な PC 操作ができること。

訓練詳細(例)：

- システム構成 ■開発プロセス ■機能要件/非機能要件
- アルゴリズムの基本 ■フローチャートの書き方
- 集計 ■集計アルゴリズムの演習
- 探索(逐次探索、2分探索) ■探索アルゴリズムの演習課題
- 整列(基本選択法、単純交換法、単純挿入法) ■整列の演習課題
- マッチング ■マッチングの演習課題
- グループトータル ■グループトータルの演習課題
- 文字列操作 ■文字列操作の演習課題

推奨訓練時間： 24 時間

C 言語基礎

狙い： 初学者をターゲットに理解しやすい項目から次第に深い内容に入っていく。
アルゴリズムを学習することで、文法上の知識だけでなく、プログラミングに必要な考え方を学習する。

訓練目標： 1. アルゴリズム基礎、リスト、スタック、キュー
2. 変数、ビットシフト、基本制御構文
3. 配列、ポインタ、文字列、構造体
4. 自作関数
5. メモリの動的確保、標準ライブラリ関数

前提知識： 基本的なコンピュータ知識があること。一般的な PC 操作ができること

訓練詳細(例)：

- アルゴリズム基礎 ■リスト ■スタック ■キュー
- 変数 ■記憶クラス ■型変換 ■定数 ■マクロ定義
- ビットシフト
- 基本制御構文(条件分岐、繰り返し) ■配列 ■自作関数
- プロトタイプ宣言 ■ポインタ ■文字列
- 構造体 ■メモリの動的確保 ■標準ライブラリ関数

推奨訓練時間： 32 時間

C 言語プログラミング

狙い： C 言語の文法を一通り学習し終えた段階から、応用プログラムを作成する。

訓練目標： 1. キーボード入力、画面表示を見に付ける
2. 構造体の利用と関数化手法を身につける
3. ファイルの扱いを見に付ける

前提知識： C 言語の文法を隔週していること

訓練詳細(例)： ※全てプログラム作成

- キーボード入力と合計加算
- 引数のポインタ渡し
- 高度な演算子の利用
- ビットフィールド
- 関数ポインタ
- ファイル IO
- 分割コンパイル
- 棒グラフ
- 構造体のソートと表示
- ファイルの読み込み
- 単語数カウント
- チェーン構造によるキュー管理(自己参照構造体を使用)

推奨訓練時間： 56 時間

組込みシステム基礎

狙い： 組込みシステムの概要、用語の定義などから始めて、システム開発方法、特徴などを初歩から学ぶ。

訓練目標： 1. 組込みシステムの特徴を理解する
2. 組込みシステムの開発方法を理解する
3. 組込みシステムで使うマイコンの利用方法を理解する

前提知識： 基礎的なコンピュータ知識があること

訓練詳細(例)：

- 組込みシステムの特徴
- マイコンとは
- マイコンで取り扱うデータ
- マイコンの要素
- マイコンを構成するしくみ
- マイコンシステムの開発環境
- マイコン制御プログラムのしくみ
- マイコンシステムの周辺機器

推奨訓練時間： 16 時間

組込みプログラミング

狙い： C 言語を使った組込みプログラム作成を学ぶ。また、C 言語の理解に必須であるアセンブリ言語についても簡単に学ぶ。

訓練目標： 1. アセンブリ言語を理解する
2. C 言語を使ったマイコン内蔵 I/O 利用方法を理解する
3. C 言語を使った周辺 I/O 利用方法を理解する

前提知識： C 言語でプログラミングができること

訓練詳細(例)：

- 組込みシステムの特徴 ■アセンブリ言語を使ったプログラム実習
- アセンブリ言語を使ったプログラム実習(LED 表示、SW 入力と LED 表示)
- サブルーチン ■組込みで使う C 言語の特徴
- C 言語を使用した組込みプログラミング実習
- ☐割込み ☐タイマによる LED 点滅 ☐タイマ割込み
- ☐LCD 表示 ☐PWM 制御 ☐AD 変換 ☐UART

推奨訓練時間： 56 時間

リアルタイム OS 基礎

狙い： 組込みシステムで利用されるリアルタイム OS について、概要・利用効果・利用方法・システムコールを学ぶ。

訓練目標： 1. リアルタイム OS の概要、利用効果を学習する
2. リアルタイム OS の利用方法、システムコールを学習する
3. リアルタイム OS の下でのドライバの作成方法を学習する

前提知識： 基礎的なコンピュータ知識があること

訓練詳細(例)：

- リアルタイム OS の概要 ■リアルタイムシステム ■マルチタスク
- リアルタイム OS の機能 ■タスク機動演習
- タスクスケジューリング ■優先順位 ■マルチタスク起動の演習
- タスク関連システムコールの演習
- セマフォ、ミューテックス関連システムコールの実習
- イベントフラグ、データキュー、メッセージバッファ、メールボックス
- 周期ハンドラを使ったシステムの作成 ■メモリプール
- 割込みハンドラ、割込みサービスルーチン
- RTOS プログラミングの注意 ■デバイスドライバの作成
- マルチタスクデバッグとソフトウェア検証

推奨訓練時間： 24 時間

ソフトウェア設計手法

狙い： 組込みソフトウェアの基本的な開発手順や手法、オブジェクト指向設計について学習し、UML モデリング・状態遷移表設計手法の演習を行い、理解を深める。

訓練目標： 1. 組込みソフトウェア開発手順と考え方を習得する
2. 設計手法を習得する
3. UML の概要、利用方法を習得する
4. 状態遷移表の概要、利用方法を習得する

前提知識： C 言語でプログラミングができること

訓練詳細(例)：

- 組込みソフトウェア開発プロセス ■オブジェクト指向設計
- UML 入門 ■UML モデリング演習
- 状態遷移設計手法入門 ■状態遷移図と状態遷移表
- 状態遷移表によるモデリング ■拡張階層化状態遷移表
- 状態遷移設計演習 ■応用演習

推奨訓練時間： 24 時間

プロジェクト型演習

狙い： 3～4 名のチーム編成で実際の開発を組込みソフトウェア開発プロセスの順を追って模擬体験する。成功・失敗体験から多くのことを学んでもらう。

訓練目標： 1. 高効率で高品質な開発に向けた上流工程の重要性を体得する
2. 開発の重要ポイント、チームで開発することのむずかしさを体得する。
(仕様書、モデリング、コミュニケーション等)
3. ハードウェアの使用や特性について十分に理解することや、
想定外のことに対応しなければならないことを体得する

前提知識： C 言語でプログラミングができること

訓練詳細(例)：

※講義はポイントのみ。問題点の解決はチーム主体で行い、講師がアドバイス

- 概要説明 ■要求仕様の提示 ■要求仕様分析
- システム概要検討 ■仕様書作成 ■仕様レビュー
- 構造化設計 ■設計書作成 ■設計レビュー
- システムのモデリング ■モデルレビュー ■詳細設計
- プログラミング ■単体テスト ■プログラミング ■結合テスト
- システムテスト ■プレゼンテーション準備 プレゼンテーション

推奨訓練時間： 72 時間 (8 時間 x 9 日間)