

# EAPOL-<sup>TM</sup> Trainer

Embedded Application development Trainer

## BASIC



サンプル版

### 注意事項(必ず確認してください)

・本書は著作権上の保護を受けています。本書の一部あるいは全部について、株式会社アイ・エル・シーから文書による許諾を得ずに、いかなる方法においても無断で複写、複製することは禁じられています。

また、本書付属の CD-ROM(内部のソフトウェアなどを含みます)のご使用にあたり、以下の点に注意してください。

- ・CD-ROM は、特定のコンピュータでのみ使用可能です。
- ・CD-ROM に含まれるソフトウェアによって生成されたソースコードおよびソースコードを利用し作成した実行モジュールを、商用利用を目的に使用することはできません。
- ・株式会社アイ・エル・シーは、CD-ROM 内のソフトウェアのご使用に関する Q&A 等のサポートは原則行いません。
- ・CD-ROM の著作権は株式会社アイ・エル・シーが有するものであり、日本国著作権法により保護されています。
- ・お客様は、CD-ROM を複製することはできません。
- ・お客様は、株式会社アイ・エル・シーに無断で、第三者に対する CD-ROM の転載、配布、公開、公衆送信、譲渡、貸与、使用許諾その他一切の行為を行うことはできません。
- ・お客様は CD-ROM 内のソフトウェアについてリバース・エンジニアリング、逆コンパイル、逆アンセンブルすることはできません。
- ・株式会社アイ・エル・シーは、いかなる場合においても、CD-ROM の使用または使用不可能から生ずる損害に関しての責任の一切を負わないものとします。
- ・CD-ROM の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、株式会社アイ・エル・シーは一切その責を負いません。
- ・本書の内容に基づいて作成したアプリケーション及びサンプル提供アプリケーションを、商用利用を目的に使用することはできません。

本書に登場する製品名などは、一般に、各社の登録商標、または商標です。

なお、本文中に TM, ®, ©マークなどは特に明記しておりません。

# 目次

|                             |    |
|-----------------------------|----|
| はじめに                        | 1  |
| 第 1 章    アプリケーション仕様概要       | 9  |
| 1.1    機能一覧                 | 10 |
| 1.2    画面仕様                 | 11 |
| 1.2.1    画面一覧               | 11 |
| 第 2 章    ソフトウェア設計           | 13 |
| 2.1    ソフトウェア設計とは           | 14 |
| 2.2    ソフトウェア設計の進め方         | 15 |
| 2.2.1    全機能の抽出と細分化         | 15 |
| 2.2.2    表示処理のグループ化         | 17 |
| 2.2.3    環境依存処理のグループ化       | 19 |
| 2.2.4    プログラム構造の設計         | 21 |
| 2.2.5    モジュール間連携の設計        | 24 |
| 2.2.6    モジュールの階層設計         | 25 |
| 2.2.7    モジュール一覧            | 27 |
| 2.3    プログラム作成のテクニック        | 29 |
| 2.3.1    ヘッダファイルの多重インクルード防止 | 29 |
| 2.3.2    モジュール構造体の使用        | 31 |
| 2.3.3    データと処理のカプセル化       | 32 |
| 2.3.4    関数名称の統一            | 34 |
| 第 3 章    プログラムの実装           | 35 |
| 3.1    本章の目的                | 36 |
| 3.2    フォルダ構成               | 38 |
| 3.2.1    プロジェクトファイルフォルダ構成   | 38 |
| 3.2.2    ソースコードフォルダ構成       | 38 |
| 3.3    GENSKETCH3 の起動       | 39 |
| 3.4    時計画面の作成              | 40 |
| 3.4.1    目標                 | 40 |
| 3.4.2    何も表示しない画面の動作確認     | 41 |
| 3.4.3    時計部品の配置            | 47 |
| 3.4.4    時計機能の実装            | 49 |
| 3.4.5    現在時刻の表示            | 60 |

|       |                      |     |
|-------|----------------------|-----|
| 3.5   | 時計設定画面の作成            | 65  |
| 3.5.1 | 目標                   | 65  |
| 3.5.2 | 時計設定画面部品の配置          | 66  |
| 3.5.3 | 時計設定機能の実装            | 68  |
| 3.6   | アラーム時刻設定画面の作成        | 94  |
| 3.6.1 | 目標                   | 94  |
| 3.6.2 | アラーム時刻設定部品の配置        | 95  |
| 3.6.3 | アラーム時刻設定機能の実装        | 97  |
| 3.7   | アラーム発生画面の作成          | 117 |
| 3.7.1 | 目標                   | 117 |
| 3.7.2 | アラーム発生画面部品の配置        | 118 |
| 3.7.3 | アラーム発生画面表示機能の実装      | 119 |
| 3.8   | 壁紙表示切換え機能の作成         | 130 |
| 3.8.1 | 目標                   | 130 |
| 3.8.2 | 壁紙表示切換え機能の実装         | 131 |
| 第 4 章 | アプリケーション仕様詳細         | 149 |
| 4.1   | 機能一覧                 | 150 |
| 4.2   | 画面仕様                 | 151 |
| 4.2.1 | 画面一覧                 | 151 |
| 4.2.2 | 時計画面                 | 152 |
| 4.2.3 | アラーム時刻設定画面           | 153 |
| 4.2.4 | 時計設定画面               | 154 |
| 4.2.5 | 時刻書式設定画面             | 155 |
| 4.2.6 | サマータイム設定画面           | 156 |
| 4.2.7 | 壁紙設定画面               | 157 |
| 4.2.8 | アラーム音設定画面            | 158 |
| 4.2.9 | アラーム発生画面             | 159 |
| 4.3   | データ仕様                | 160 |
| 4.3.1 | ユーザ登録壁紙              | 160 |
| 4.3.2 | ユーザ登録アラーム音声          | 162 |
| 第 5 章 | アプリケーション実行手順         | 165 |
| 5.1   | プログラム開発環境について        | 166 |
| 5.2   | Microsoft Windows 環境 | 166 |
| 5.2.1 | プロジェクトファイルの作成        | 166 |

|       |                                    |     |
|-------|------------------------------------|-----|
| 5.2.2 | ソースコードの登録                          | 167 |
| 5.2.3 | プロジェクトプロパティの設定                     | 169 |
| 5.3   | Android 環境                         | 177 |
| 5.3.1 | 概要                                 | 177 |
| 5.3.2 | Android NDK を使用した C 言語ソースコードのコンパイル | 177 |
| 5.3.3 | Android SDK を使用したアプリケーションの作成       | 182 |
| 第 6 章 | GUI 画面の作り方                         | 187 |
| 6.1   | 画面設計アプリケーションの起動                    | 188 |
| 6.2   | プロジェクトの新規作成                        | 188 |
| 6.3   | 共通リソースの登録                          | 192 |
| 6.3.1 | リソースへのアクセス                         | 192 |
| 6.3.2 | 画像リソースの登録                          | 193 |
| 6.3.3 | フォントリソースの登録                        | 196 |
| 6.3.4 | 文字列リソースの登録                         | 198 |
| 6.3.5 | ビットマップテキストリソースの登録                  | 201 |
| 6.4   | スクリーンの設定                           | 206 |
| 6.4.1 | スクリーンプロパティの設定                      | 206 |
| 6.5   | レイヤの設定                             | 207 |
| 6.5.1 | レイヤプロパティの設定                        | 207 |
| 6.6   | パネル/ウィンドウの作成                       | 209 |
| 6.6.1 | パネル/ウィンドウの新規作成                     | 209 |
| 6.6.2 | 時計画面の作成                            | 210 |
| 6.6.3 | 時計設定画面の作成                          | 219 |
| 6.6.4 | アラーム時刻設定画面の作成                      | 229 |
| 6.6.5 | 項目設定画面の作成                          | 235 |
| 6.6.6 | アラーム発生画面の作成                        | 240 |
| 6.7   | ユーザ定義イベントの設定                       | 242 |
| 6.7.1 | イベントハンドラの設定                        | 242 |
| 第 7 章 | 用語集                                | 246 |

# はじめに

---

本書は、組込み開発現場での開発経験に基づいた実戦的な開発ノウハウを組込みアプリケーション開発者の方に対して提供するものです。

シリーズ構成としては下記を予定しており、本書は「基礎編」となります。

## ■基礎編：EAPL-Trainer BASIC

一般的なアプリケーション開発と組込みアプリケーション開発の違いと、組込みアプリケーション開発の基本的な設計・実装手法を解説します。

組込みアプリケーション開発の初心者を対象としており、組込みアプリケーション開発の基礎が習得できます。

## ■実装応用編：EAPL-Trainer APPLICATION

組込みアプリケーション開発で使用する様々な実装手法の目的に応じた使い分けについて解説します。

組込みアプリケーション開発のプログラミングを行う方を対象としており、様々な実装手法について習得できます。

## ■設計応用編：EAPL-Trainer CHALLENGE

組込みアプリケーション開発において、状態遷移の概念を取り入れた効率の良い設計手法を解説します。

組込みアプリケーション開発の設計を行う方を対象としており、状態遷移設計の手法について習得できます。

## (1) 本書の対象読者

本書は C 言語を使って組込み機器アプリケーションの開発を行うための基本的なノウハウをまとめたものです。

読者の知識としては C 言語を使用して PC 上でプログラムを作成できるレベルを想定しています。また、オブジェクト指向の基本的な考え方を理解していると、より理解がしやすいと思いますが必須ではありません。

次のような人を読者として想定しています。

- これから組込み機器で動作するプログラムを開発するが、どのようにして設計・実装してよいかわからない人
- 既に組込み機器で動作するアプリケーションを開発しているが、より効率よく開発を行いたいと考えている人
- これから組込みアプリケーションの開発者を目指そうとしている人

SAMPLE

## (2) 組み込み機器アプリケーション開発の難しさ

組み込み機器アプリケーションの開発と一般的な PC 上のアプリケーションの開発は何が違うのでしょうか？

組み込み機器の開発と、一般的な PC 上での開発は以下のような点が異なります。

- オブジェクト指向言語が使えないことが多い

組み込み機器の開発ではプログラムの大きさの制約が PC に比べて厳しい場合が多く、例えば PC 上で動作するプログラムの 1/10 の大きさでプログラムを作らなければならないこともあります。

C++/Java/C#などのオブジェクト指向言語を使ってプログラムを作成すると、できあがったプログラムの大きさが C 言語などを使った場合に比べて大きくなります。また、動作に必要なメモリサイズも大きくなります。そのため、開発にオブジェクト指向言語が使えないことが多く、C 言語を使うことになります。

C 言語はオブジェクト指向言語ではありませんが、オブジェクト指向の考え方を取り入れた設計やプログラム作成を行うことによって開発効率やメンテナンス性を高めることが可能となります。本書ではその基本的な手法について説明します。

- PC 上の開発に比べてデバッグ手段が貧弱であることが多い

PC 上でアプリケーション開発を行う場合、Microsoft の Visual Studio のような統合開発環境が使えますので、プログラムの動きに問題が起きた場合には様々なデバッグ機能を使って問題を解決することができます。

しかし、組み込み機器の開発ではそのような統合開発環境を使えないことが多いです。その場合、開発環境のデバッグ機能にできるだけ頼らずにプログラムの問題を解決しなくてはなりません。本書では、問題発生時の追跡をできるだけ簡単に行うための設計の手法について説明します。

- 一度作ったプログラムを他の機器に移植することがある

組み込み機器の場合、ある機器向けに開発したアプリケーションを別の機器(後継機種など)用に移植することがあります。

新しい機器に移植を行うということは、プログラムを新しい機器に合わせて修正することになります。あらかじめ、移植することを考えて設計を行わないと、プログラム全体を修正することになって、移植に非常に手間がかかることもあります。

アプリケーションの処理は、どんな機器でも同じ処理でよい共通部分と、機器によって処理を変える必要のある機器依存部分に分けられることが多いです。共通部分と機器依存部分を明確に切り離して設計することによって、移植するときの手間を最小限に抑えることができます。本書では、そのような設計を行うための方針や手法について説明します。

以上のような違いに注意して組込み機器の開発を進める必要がありますが、そのような観点で書かれた書籍は一般的なアプリケーション開発の書籍に比べて数が少ないというのが現状です。

本書は、組込み開発現場での様々な開発経験に基づいた実戦的な設計とプログラム作成の手法を具体的な例題を使って説明します。

組込み機器開発の初心者が実戦的なノウハウの基礎を身につけて簡単な組込みアプリケーションであれば自力で開発できることを目的として本書を作成しています。

SAMPLE

### (3) 本書の構成と使い方

本書はサンプルアプリケーションを題材として組込みアプリケーション開発のノウハウを順番に説明します。

最後まで読み進めることにより 1 つのデジタル時計アプリケーションを完成させることができます。

本書で作成するアプリケーションは複数の機器で動作することを念頭において設計していますので、Microsoft Windows 環境と Android 端末の両方で動作します。もし、Android 端末をお持ちであれば Microsoft Windows と Android 端末の両方の環境で同じアプリケーションの動作を確認することができます。

本書の内容は下記のとおりです。

#### 第 1 章 アプリケーション仕様

題材とするアプリケーションの画面構成や機能について説明します。作成するアプリケーションの機能や使い方を理解します。

アプリケーションの詳細な仕様については第 4 章で説明します。

#### 第 2 章 ソフトウェア設計

第 1 章のアプリケーション仕様に基づくソフトウェア設計の方針について説明します。

具体的なソフトウェア設計の内容を示しながら、ソフトウェア設計を行う上での基本的な考え方や注意すべき点について説明します。

また、オブジェクト指向を考慮した設計や移植性を考慮した設計について説明します。

#### 第 3 章 プログラムの実装

第 2 章のソフトウェア設計に基づくプログラム作成の手法について説明します。

アプリケーションの画面を 1 つずつ作成しながら順番にプログラムを作成していきます。1 画面ずつ順番に作成したサンプルプロジェクトを同梱していますので、それらを使いながら順番にプログラムの動作を確認することができます。

#### 第 4 章 アプリケーション詳細仕様

本書で題材とするアプリケーションの画面構成や機能に関する詳細な仕様を説明します。

#### 第 5 章 アプリケーション実行手順

Microsoft Windows 環境でプログラムを作成・実行するための設定を説明します。同梱のサンプルプロジェクトではこれらの設定は完了している状態ですが、最初からプロジェクトファイルを作成するときには本章の内容を参考にしてください。

#### 第 6 章 GUI 画面の作り方

本キットに同梱されている GUI 画面設計ツールを使って実際に GUI 画面を作成する手順を説明します。同梱のサンプルプロジェクトでは GUI 画面が作成済みの状態ですが、最初から GUI 画面を作成するときには本章の内容を参考にしてください。

#### (4) 添付 CD-ROM について

本書に添付されている CD-ROM には以下が含まれます。

| 項目           | 内容                                   |
|--------------|--------------------------------------|
| GUI 画面作成ツール  | 3 章で本ツールを使い、GUI 画面の作画とソースコード生成を行います。 |
| サンプルアプリケーション | 本書に沿って作成されるアプリケーションです。               |

これらを使用するには、CD-ROM 内のインストーラから以下の手順でセットアップを行います。

1. Windows に管理者権限でログオンします。
2. CD-ROM をパソコンの CD-ROM ドライブにセットします。
3. CD-ROM 内の「Setup.exe」を起動します。
4. 表示されるインストールウィザードに従い、インストールを実行します。

インストール完了後、Windows のプログラムグループ[ILC]を選択し、[EAPL-Trainer]-[Embedded G UI]から各ツールを起動します。

インストールに必要なシステム要件

| 要件      | 内容                                                                                          |
|---------|---------------------------------------------------------------------------------------------|
| OS      | Windows XP Professional (SP2 以上)、<br>Windows 7 Professional (32 ビット版、64 ビット版)               |
| ハードウェア  | PC/AT 互換機                                                                                   |
| CPU     | 1.0GHz 以上の Intel PentiumⅢ相当以上のプロセッサ                                                         |
| メモリ     | ・ Windows XP : 512MB 以上<br>・ Windows 7 (32 ビット版) : 1GB 以上<br>・ Windows 7 (64 ビット版) : 2GB 以上 |
| ハードディスク | 700MB 以上の空き容量 (インストール用およびプロジェクトデータ用)                                                        |
| ディスプレイ  | XGA (1024×768) 以上推奨、32 ビットカラー以上                                                             |
| ディスク装置  | CD-ROM ドライブ                                                                                 |

## (5) 本書の利用に必要な開発環境

- GENWARE3

Microsoft Windows 上で動作する組込み GUI 統合開発環境です。

付属 CD-ROM の“GENWARE3”フォルダ内にインストールガイドが入っています。

インストールの方法についてはインストールガイド (GENWARE3 インストールガイド.pdf) を参照ください。

- Visual C++ 2008 Express Edition

Microsoft Windows 上で C 言語プログラムをコンパイルするためのコンパイラです。

下記の URL からダウンロードから可能です (2011 年 8 月 現在)。

<http://www.microsoft.com/japan/msdn/vstudio/2008/product/express/>

本書は上記のコンパイラを使用することを前提として記載していますが、提供されなくなる可能性があります。

その場合は、その時点での最新のバージョンを使用してください。

SAMPLE

SAMPLE

# 第1章 アプリケーション仕様概要

本章では、本書で開発するデジタル時計アプリケーションの外観、画面の切換え手順などを説明します。

SAMPLE

# 1.1 機能一覧

本書で開発するサンプルアプリケーションはアラーム機能付き時計アプリケーションです。  
時計の背景表示としてユーザが用意した画像ファイルを使用したり、アラーム発生時に鳴る音声としてユーザが用意した音声ファイルを使うことができます。

表 1-1 サンプルアプリケーションの機能一覧

| 項目     | 概要                                                                                               |
|--------|--------------------------------------------------------------------------------------------------|
| 時計機能   | 現在の日付と時刻を表示します。<br>時刻の表示形式は 12 時間制表示と 24 時間制表示が可能です。<br>サマータイム表示が可能です。                           |
| アラーム機能 | アラーム時刻を設定することができます。<br>設定した時刻にアラーム音を鳴らします。                                                       |
| 時計設定機能 | 12 時間制と 24 時間制の切り換えができます。<br>サマータイム表示の切り換えができます。<br>時計画面の背景表示を選択することができます。<br>アラーム音を選択することができます。 |

## 1.2 画面仕様

### 1.2.1 画面一覧

時計アプリケーションは時計画面を中心に様々な画面に切り換わります。

画面の切換えはタッチ操作によって実行します。各画面での操作方法の詳細は「第4章 アプリケーション仕様詳細」を参照ください。

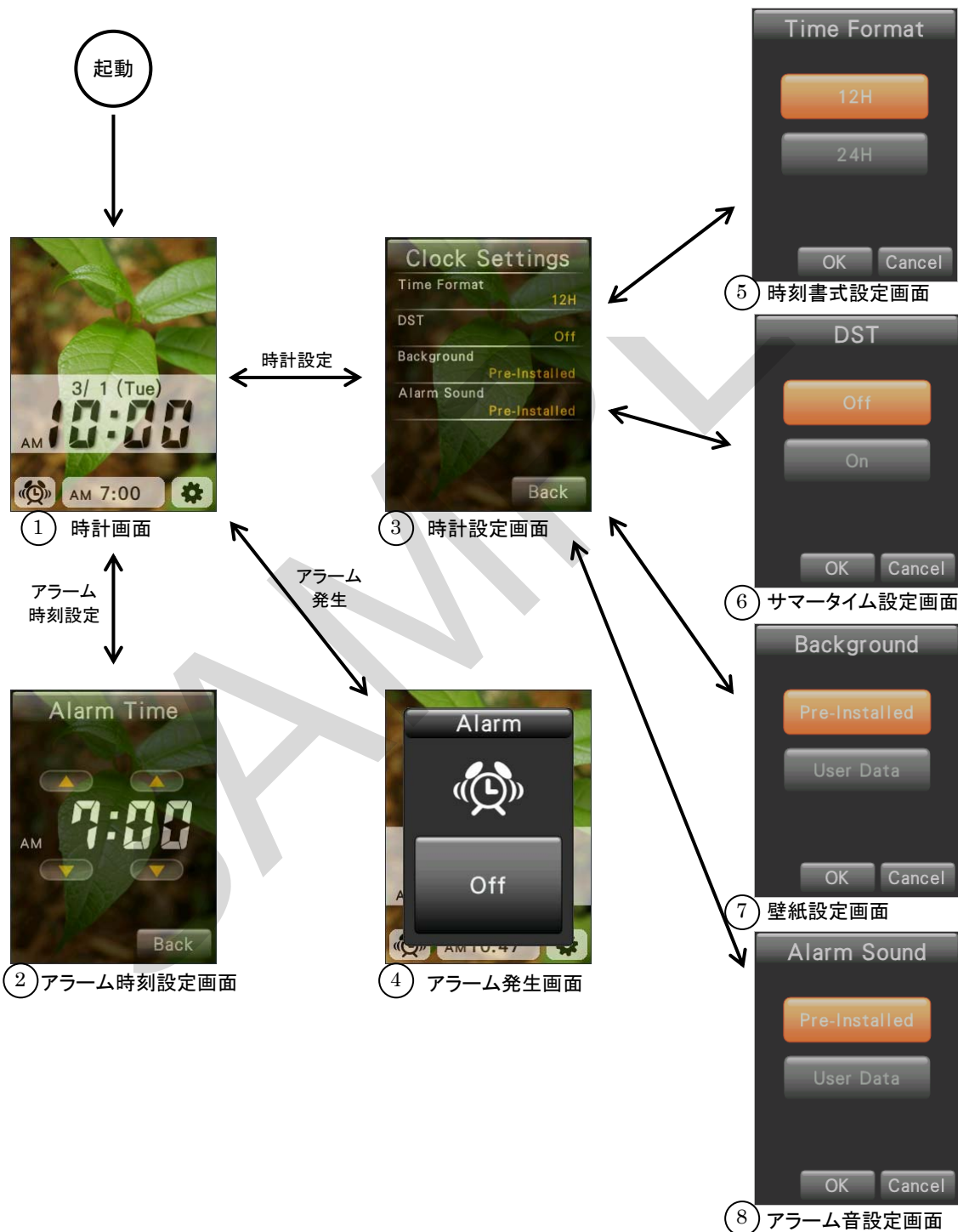


図 1-1 時計アプリケーションの画面一覧

SAMPLE

## 第2章 ソフトウェア設計

---

2

本章では、効率よく組込みアプリケーションを開発するためのソフトウェア設計の方針を説明します。また、ソフトウェア設計の具体的な例として本書で作成するサンプルアプリケーションの設計内容を説明します。

SAMPLE

## 2.1 ソフトウェア設計とは

アプリケーションを作成するためには、実際にソースコードを書き始める前にプログラムの構造を考えて決める必要があります。この「構造を考えて決める」という行為をソフトウェア設計と呼びます。

ソフトウェア設計は非常に重要な行為です。ソフトウェア設計の出来の良し悪しによって、プログラムの作りやすさや修正のしやすさが決まると言っても過言ではありません。

ソフトウェア設計の方針や手法に関しては様々なやり方がありますが、本章では組込みアプリケーション開発に適したソフトウェア設計の手法に関して説明します。

SAMPLE

## 2.2 ソフトウェア設計の進め方

### 2.2.1 全機能の抽出と細分化

まず、アプリケーションで実現すべき機能を全て列挙します。

機能の見落としがあると、後から機能を追加するためにプログラムの構造を見直すことになったり、そのまま機能を作り忘れてしまうという危険性がありますので最初に行います。

本書のサンプルアプリケーションの場合、アプリケーション仕様(第1章や第4章に記載)を基に全機能の抽出を行います。

アプリケーション仕様を基に大まかな機能構成を検討すると時計機能・アラーム機能・時計設定機能の3つの機能から成り立つと考えることができます。

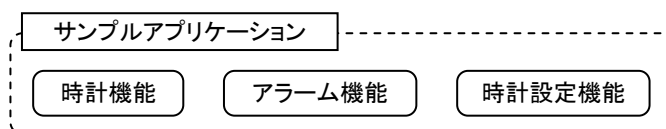


図 2-1 サンプルアプリケーションの機能一覧

これらの機能を、プログラムによって実現するためにはもっと細かい単位に分割する必要があります。アプリケーションの動作としては一つの機能であっても内部では細かい部品の組み合わせで実現します。

部品とはプログラムの構成要素のことです。例えば、1台の自動車がたくさんの部品から構成されているように、アプリケーションプログラムもたくさんの部品から構成されます。

細かい部品に分ける理由は主に2つあります。

- 部品の大きさが小さいほうが問題が起こりにくい

プログラムの作成は部品単位で行うので、大きさが小さい方が作りやすくて動作確認も簡単になります。結果として問題が起こりにくくて質の高いプログラムになります。

- 似ている部品はグループにまとめた方がよい

細かい部品に分けると、同じような機能を持つ部品が複数できることがあります。それらの部品を一箇所にまとめた方がプログラムを共通化することができます。

同じような機能のプログラムをまとめずに複数の場所に散在させているとプログラム作成の手間が無駄です。また、後で修正する場合に全てのプログラムを漏らさず書き直す必要があり、手間もかかりますし書き直し漏れが発生する危険性があります。

SAMPLE

サンプル版はここまでとなります。