



MONOist > 組み込み開発 > 状態遷移表による設計手法(5):状態遷移表からの...

状態遷移表による設計手法(5):

状態遷移表からの実装 (1/3)

2012年11月07日 10時00分 公開

「塚田 雄一 キヤッツ, MONOist」

印刷

通知

3

Share

7

はじめに

組み込みソフトウェアが抱える一番の課題は「**設計品質の向上**」です。[本連載](#)の主役「**状態遷移表**」であれば、“イベント”と“状態”の全ての組み合わせを捉えることができるため、「モレ」「ヌケ」のない品質の良い設計が可能です。そして、不具合発生による手戻りコストの削減や開発効率の向上にも役立ちます。

こうした理由から、組み込みソフトウェア開発の世界では、長年、**状態遷移系モデル**で設計が行われています。

前回は、“状態遷移表を使用した設計モデル”をテーマに、**拡張階層化状態遷移表**を用いた設計手法を紹介しました。ソフトウェア開発とは、当然のことながら、設計書の作成がゴールではありません。最終的に、プログラミング(モデルからの自動コード生成を含む)を行い、ソフトウェアを完成させることがゴールです。つまり、**設計書は、プログラミングを行う目的のために作成するもの**と言い換えることもできます。ということで、今回は、「**状態遷移表からの実装**」について解説します。

なお、本連載では以下の6つのテーマを順番にお届けしていきます。

1. (連載第1回):状態遷移表設計手法の概要
2. (連載第2回):なぜ状態遷移表を使うと、品質の良い開発ができるのか
3. (連載第3回):状態遷移表を使用した要求分析モデル
4. (連載第4回):状態遷移表を使用した設計モデル(拡張階層化状態遷移表)
5. 状態遷移表からの実装
6. 状態遷移表を使用したテスト手法

フローチャートとは

今回のテーマ、状態遷移表からの実装の本題に入る前に、“**プログラミングを行うための表現方法**”について確認しておきましょう。

プログラミングの世界には、C、C++、Javaなど、さまざまな言語が存在します。これと同じように、プログラムの構造や処理の流れを表現するものとして、**フローチャート**や、PAD、HCPチャートといった構造化プログラミングを行うための**構造化チャート**など、実に多くの表記手法があります。

今回は、最も基本的な手法であり、単純な表記で構成されるフローチャートを用いながら、解説を進めていきます。

カスタム検索



スポンサーからのお知らせ

- PR -

> **【5月29日 東京開催】MONOist 主催セミナー**
 失われつつある「日本品質」という強み、
 「攻めの品質管理」で生まれる新たな価値

Special Contents

- PR -



「ねじレス化」が生み出す価値、盤製作全体の効率化を目指す制御・配電盤革新



IoTのPoC段階はもう終わり、工場のスマート化が本格的に加速へ



次世代Power over Ethernet規格「PoE++」対応機器を実現するチップセット



IoT時代の安全と安心を確保するワンストップの組み込みプラットフォームとは **New!**



日本の製造業が第4次産業革命の波に乗るために必要な“3つのIoT活用”とは



デンソーが次世代車載コックピット
への採用を決めたフラッシュメモリ
とは? **New!**



スマートファクトリー化でCC-Link IE TSNが果たすべき役割

フローチャート(Flow Chart)とは、“流れ図”または“流れ作業図”のことです。各工程(処理)を示す部品は“JIS X 0121:1986”により規格化されており、形・流れる方向などが統一化されています。端子からスタートし、処理の流れは、→で表現します。各処理は、□で表現し、選択、反復などの条件部分は、◇で表現します(図1)。

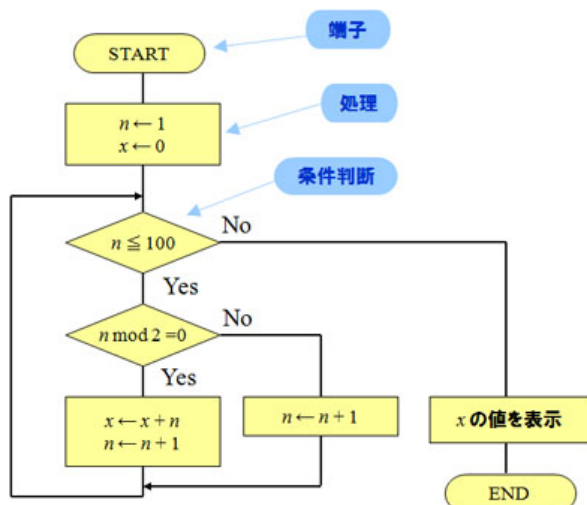


図1 フローチャートとは

キッチンタイマーの基本仕様から直接実装した場合

今回の目的は、状態遷移表からの実装プロセスを紹介することですが、その前に、“基本仕様から直接実装した場合”を想定したフローチャートを作成します。ここで作成したフローチャートは、後ほど、“状態遷移表から実装した場合”のフローチャートと比較します。

図2に、連載第3回「状態遷移表を使用した要求分析モデル」で掲載したキッチンタイマーの基本仕様を示します。

基本仕様

- ・ 時間設定ボタン(分、10秒、クリア)により、カウントダウン時間設定を行う
- ・ スタート/ストップボタンが押された場合、時間設定完了の場合、カウントダウンを開始する。もしもカウントダウン中であれば、カウントダウンを一時停止する
- ・ カウントダウンが終了した場合は、カウントダウンを停止し、アラーム音を出力する。スタート/ストップが押された場合、アラーム音を停止する



図2 キッチンタイマーの基本仕様

図2に示した基本仕様に従って、下記の手順で処理の流れをフローチャートで表現していきます。

(1)まず、「時間設定ボタン」が押された場合は、「カウントダウン時間設定」処理を行います。同時に、「スタートストップボタン」が押された場合は、「カウントダウン開始」処理を行います。これらのイベントをループで参照し続けます。



プラットフォームにならないマイクロソフトの「CASE戦略」New!



現実を超えた仮想環境へ、自動運転時代に向けた最新モデルベース開発ツール



日本の製造業が直面する課題とその解決、マイクロソフトが描く変革のシナリオ

» Special 一覧

Special Site

- PR -



【Embedded Innovations】

マイコン/アナログ/メモリ最新情報を配信中。
組み込みの最新情報をチェック



第4次産業革命をチャンスに

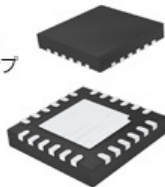
日本の製造業が直面する課題とその解決、マイクロソフトが描く変革のシナリオ

LTC6560/LTC6561 TIA
アンプは、LIDARおよび
産業用画像処理向け

出力多重化機能付き
シングルおよび4チャンネル
トランスインピーダンスアンプ

ANALOG
DEVICES
AHEAD OF WHAT'S POSSIBLE™

Digi-Key
ELECTRONICS
メーカー公認
ディストリビュータ



詳細はこちら

コーナーリンク

Windows7サポート終了 対策ナビ

Windows 7 サポート終了 対策ナビ

Windows 10 IoT

FPGA

車載ソフトウェア

組み込み開発の記事ランキング



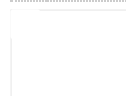
深層学習初心者向けの無料オンライン
学習資料を公開



いまさら聞けないLPWAの選び方
【2019年春版】



低速自動運転を“商品レベル”で実現
へ、ヤマハ発動機とDMPがAIで資本提
携



日本初のAIプロダクト品質保証ガイド
ライン、QA4AIコンソーシアムが発行へ



CANプロトコルを理解するための基礎
知識

(2)カウントダウンが開始された後で、「カウントダウン終了」した場合は、「カウントダウン停止&アラーム音出力」処理を行います。また、カウントダウン中で、カウントダウンが終了する前に、「スタートストップボタン」が押された場合は、「カウントダウン一時停止」処理を行い、カウントダウン処理を開始する前の「スタートストップボタン」を待っている箇所に移動します。

(3)アラーム音出力後で、「スタートストップボタン」が押された場合は、「アラーム音停止」処理を行い、時間設定を行う最初の箇所に戻ります。

フローチャートの表現方法は人それぞれなので、「これが絶対!」と一概には言えませんが、参考までに、今回作成したフローチャートを図3に示します。

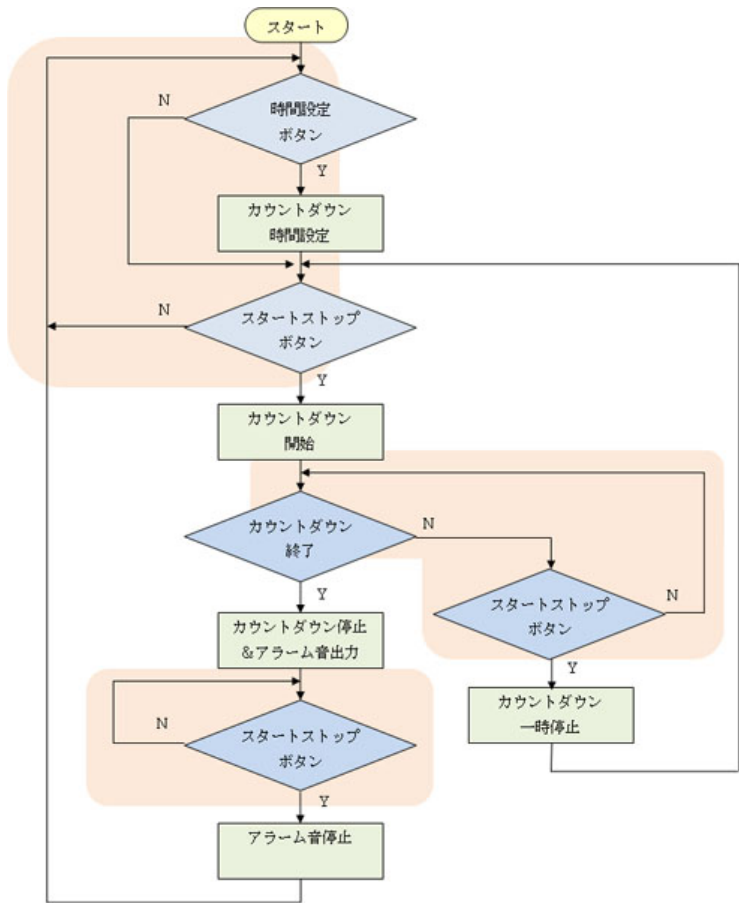


図3 キッチンタイマーの基本仕様をフローチャートで表現

基本仕様から直接作成したフローチャートの問題点

それでは、図3のフローチャートを分析していきましょう。図3のフローチャートは、状態に着目せずに、処理の流れのみに着目して作成したため、下記のような特徴があります。

- ・【特徴1】:「スタートストップボタン」を、3箇所参照している
- ・【特徴2】:処理の目的別に3つの“マイナーループ”が存在している
- ・【特徴3】:上部から下部へ“順序性”のある状態遷移に適している

図3のフローチャートでは、「スタートストップボタン」を、3箇所参照しています。同じ処理を3箇所それぞれで行う構造は、“良い構造”とはいえません。同じイベントであれば、1箇所にまとめて表現する方が、品質やメンテナンス性も良くなります。そのため、「スタートストップボタン」を1箇所参照する構造を検討する必要があります。

また、マイナーループが3つ存在しています。マイナーループは、処理の流れに従って、各イベント待ちを行っていますが、そのイベント待ちを行っているマイナーループが複数存在しています。マイナーループが複数存在すると、イベントの追加や変更を行う際に、それぞれのマイナーループで追

| | |
|--------------------------|---|
| <input type="checkbox"/> | HPCとAI性能を両立したポスト「京」のCPU、ウエハーが初公開 |
| <input type="checkbox"/> | CAN通信におけるデータ送信の仕組みとは？ |
| <input type="checkbox"/> | ポスト「京」のプロセッサ「A64FX」はArmベースながら異彩放つ重厚系 |
| <input type="checkbox"/> | スマートグラス活用ソリューションを保守点検業務に採用 |
| <input type="checkbox"/> | PythonだけでIoTのPoCを組める／グラフエンで世界最高感度の赤外線センサー |

よく読まれている編集記者コラム



「LOVOT」のプロダクトデザインから学んだ“仕事の流儀”



製造業も人ごとではない、もう1つの2025年問題



期待高まる金属3Dプリンタの本格導入、意識の壁を超えられるか

» 編集後記一覧

人気記事ランキング

- PR -

提供オートモーティブ・ジョブズ
AUTOMOTIVE JOBS



【ホンダ】HEV/PHEV用で熱効率40%超を達成したエンジン戦略



男性がつけていたら恥ずかしい「図柄入りご当地ナンバー」ランキング



45%が「勝手にブレーキをかけてくれる」と認識 自動ブレーキにまつわる誤解とリスク



【日系自動車メーカー・ADAS比較】自動運転・ADAS技術開発の最新動向



【動画で解説】研究開発の採用動向 CASE関連の中途採用が進む

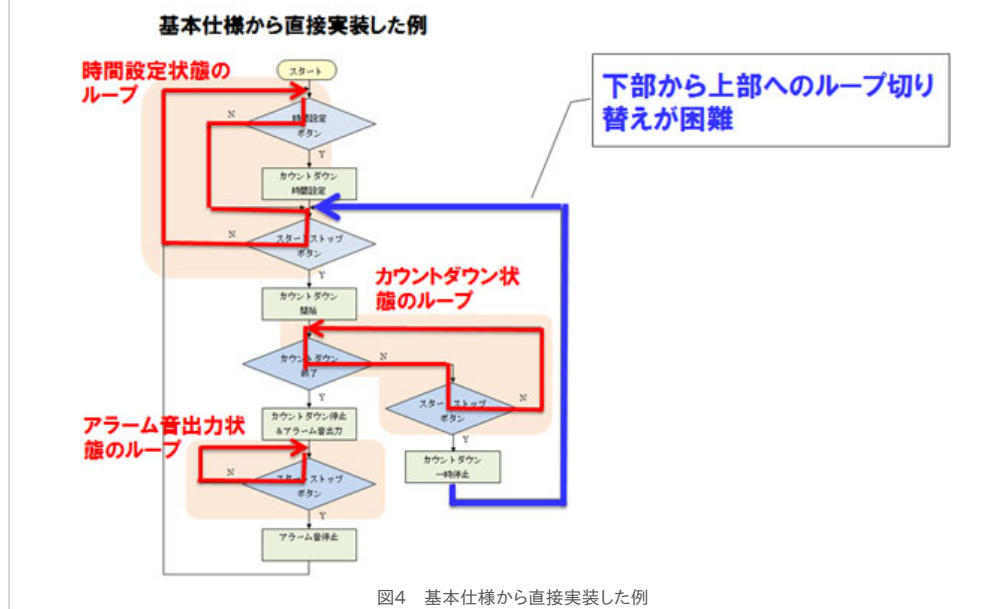
» 他の記事を見る

加・変更を検討しなければなりません。つまり、品質やメンテナンス性を低下させる要因となります。従って、マイナーループをできるだけ少なくする構造を検討するべきです。

さらに、**図3**のフローチャートは、下部のマイナーループから上部のマイナーループへと動作する必要があります。プログラムは上部から下部へと動作するため、“順序性”が明確な切り替えには適していますが、カウントダウン状態のループから時間設定状態のループへ戻る状態遷移のように、“下部から上部へと移動する構造”の実装には、状態のループを切り替えるためのフラグや変数などが必要になってしまいます。

- 以上のことから、下記のような検討課題が見えてきます。
- ・【検討課題1】:「スタートストップボタン」を1箇所参照したい
 - ・【検討課題2】:“マイナーループ”をやめて、1つのループにしたい
 - ・【検討課題3】:下部から上部へのループ切り替えにも容易に対応したい

ところで、**図3**にある3つのマイナーループが、状態遷移設計の“状態”と同じであることに気が付いた方はいますか？一番上が時間設定状態の際にループする部分です。そして、次がカウントダウン状態の際にループする部分で、一番下がアラーム音出力状態の際にループする部分です。このことから、**図3**のフローチャートは、マイナーループを使用して状態を表現しており、“マイナーループの切り替えにより、状態を切り替える構造”といえます。



実装における駆動型

それでは、状態遷移表(**図5**)からフローチャートを作成する手法を紹介します。

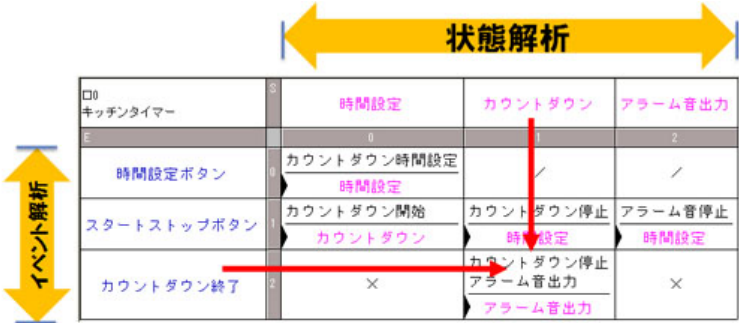
| □0 キッチンタイマー | S | 時間設定 | カウントダウン | アラーム音出力 |
|----------------|---|----------------------|---------------------------------|-----------------|
| E | | 0 | 1 | 2 |
| 時間設定ボタン | 0 | カウントダウン時間設定 時間設定 | / | / |
| スタートストップボタン | 1 | カウントダウン開始 カウントダウン | カウントダウン停止 時間設定 | アラーム音停止 時間設定 |
| カウントダウン終了 | 2 | × | カウントダウン停止 アラーム音出力 アラーム音出力 | × |

図5 状態遷移表(連載第3回「状態遷移表を使用した要求分析モデル」の最終成果物のモデルより)

図5の状態遷移表からフローチャートを作成する手法は、「イベントドリブン型(E型)」と「ステートドリブン型(S型)」の2種類があります。それぞれ、イベント主体で実装するか、状態(ステート)主

体で実装するかを指定した型になります。

イベントドリブン型は、イベント主体で実装を行うため、先にどの「イベント」が入ったかを解析し、その後、現在の「状態(ステート)」を解析して、動作させる処理を確定する型です。これに対し、ステートドリブン型は、状態(ステート)主体で実装を行うため、先に現在の「状態(ステート)」を解析し、その後、どの「イベント」が入ったかを解析して、動作させる処理を確定する型です。イベントドリブン型とステートドリブン型の特徴については後ほど詳しく説明します。



イベントドリブン型(E型): ①イベント解析→②状態解析
ステートドリブン型(S型): ①状態解析 →②イベント解析

図6 「イベントドリブン型(E型)」と「ステートドリブン型(S型)」

関連キーワード

ソフトウェア | モデルベース開発 | 設計 | 組み込み | 組み込みソフトウェア | UML | 組み込みシステム | モデリング

→ 次ページ イベントドリブン型(E型)

1 2 3 次のページへ

組み込み開発

MONOist トップ

Copyright © ITmedia, Inc. All Rights Reserved.

Special Contents

- PR -



スマートファクトリー化で
CC-Link IE TSN が果た
すべき役割



「ねじレシ化」が生み出す
価値、盤製作全体の効率
化を目指す制御・配電盤
革新



自動運転時代の自動車
開発の必需品! 最新モデ
ルベース開発ツールを一
挙紹介



日本の製造業が第4次産
業革命の波に乗るために
必要な“3つのIoT活
用”とは



コネクタ、センサーが実
現する次世代モビリティ
社会、「ホロレンズ」で体
験



IoTのPoC段階はもう終
わり、工場のスマート化が
本格的に加速へ



デンソーが次世代車載コ
ックピットへの採用を決め
たフラッシュメモリとは?
New!



自動車業界の大変革CA
SEに向けて、マイクロソ
フトが持つ“部品” **New!**