



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)。

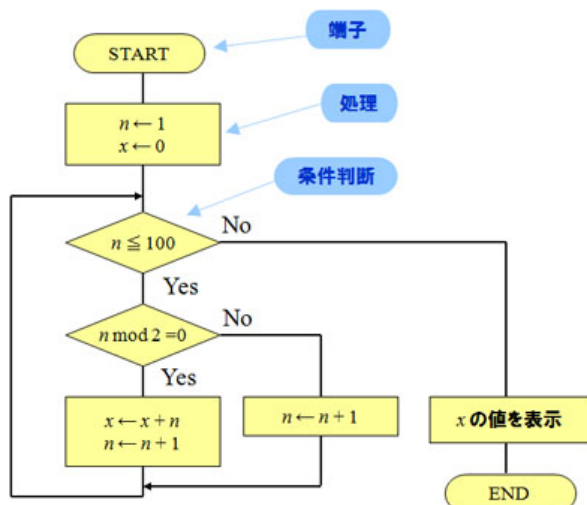


図1 フローチャートとは

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

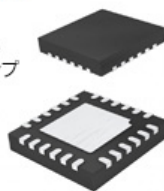
日本の製造業が直面する課題とその解決、マイクロソフトが描く変革のシナリオ

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



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

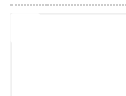
FPGA

車載ソフトウェア

組み込み開発の記事ランキング



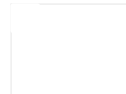
深層学習初心者向けの無料オンライン
学習資料を公開



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。

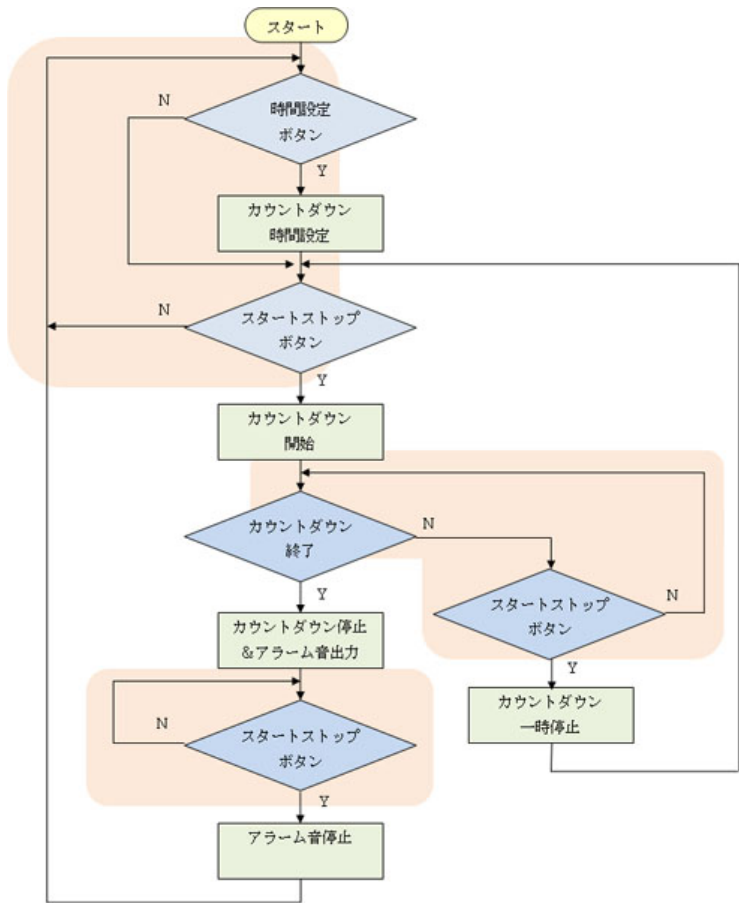


図3 キッチンタイマーの基本仕様をフローチャートで表現

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

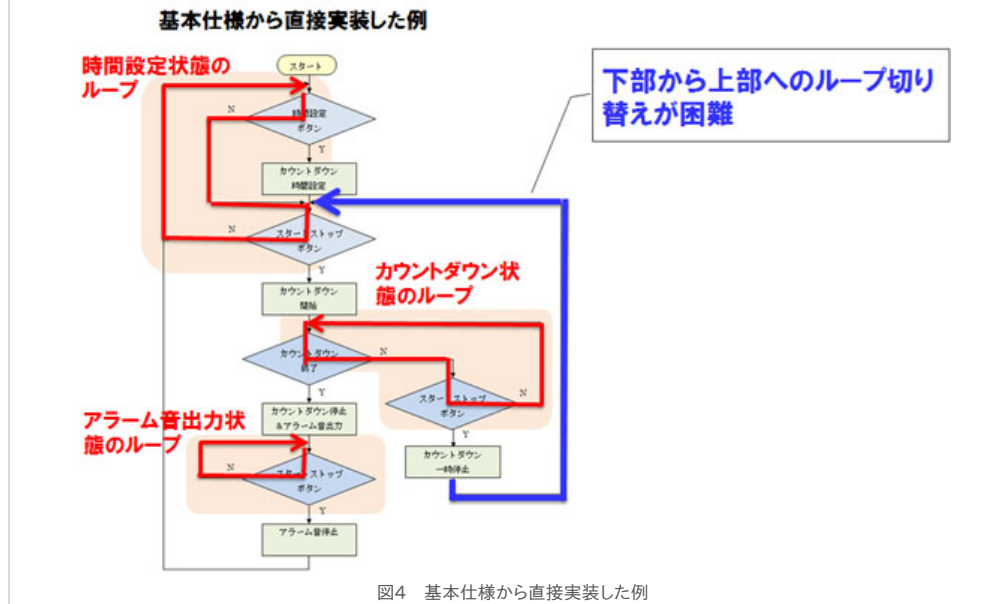
加・変更を検討しなければなりません。つまり、品質やメンテナンス性を低下させる要因となります。従って、マイナーループをできるだけ少なくする構造を検討すべきです。

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



「ねじり化」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!**

MONOist 主催セミナー

**製造業×品質、転換期を迎える
モノづくりの在り方**

開催日: 2019/5/29 (水)

会場: 野村コンファレンスプラザ日本橋

MONOist

無料

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(2/3)

2012年11月07日 10時00分 公開

[塚田 雄一 キャッツ, MONOist]

前のページへ 1 2 3 次のページへ

印刷

通知

3

Share

7

イベントドリブン型(E型)

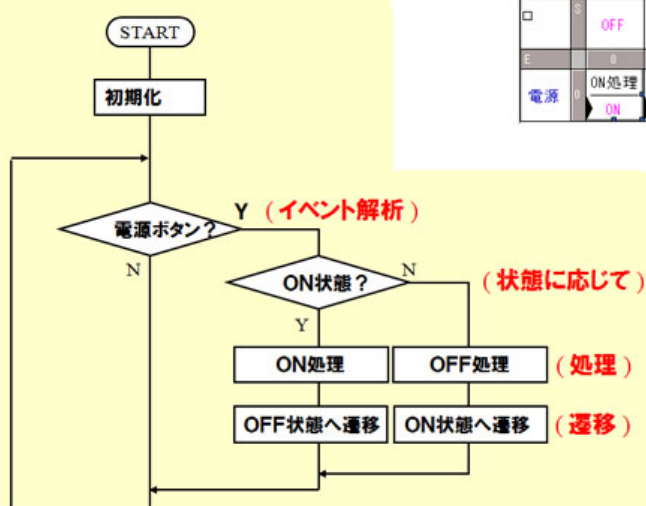
イベントドリブン型は、先にイベントを解析するため、プログラムの1箇所ですべてのイベントを解析します。そして、イベントを解析した後に、入力したイベントに関係する状態のみを解析します。つまり、各部で“必要な状態のみ解析”するということです。例えば、アクションセルが“無視”や“不可”となる状態のアクションについては解析を行いません。

主に、デバイスコントロールを行うような制御系など、入力イベントを主体として開発する分野では一般的な構造であり、多く使用されています。例えば、割り込みが発生した場合は、最初に、どのイベントが発生したかを解析することが多く、そのようなシステムにはイベントドリブン型の実装が適しています。

以下に、イベントドリブン型の特徴を記します。

- 常に全てのイベントを解析する
- 必要な状態のみを解析する
- アクションセルが“無視”や“不可”となるアクションセルの状態の解析は行わない

イベントドリブン型で実装した例



状態遷移表

	0	1
電源	ON処理	OFF処理
	OFF状態へ遷移	ON状態へ遷移

図7 イベントドリブン型(E型)で実装した例

カスタム検索

Embedded
組み込み開発

関連ニュースや会場レポートをお届け!

テクノフロンティア ものづくりエンジニアのための 技術開発促進と市場創出の場

TECHNO-FRONTIER 2019

MONOist EE Times EDN Tech Factory 詳しくはこちら

スポンサーからのお知らせ

- PR -

> 【MONOist 主催セミナー】5月29日 東京開催!

限界を迎えた現場主導の品質保証——
解決のカギと新たな「攻めの品質管理」

Special Contents

- PR -



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!**



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



コネクタ、センサーが実現する次世代モビリティ社会、「ホロレンス」で体験



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



「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

キッチンタイマーをイベントドリブン型(E型)で実装

それでは、状態遷移表からイベントドリブン型で実装したフローチャートを作成してみましょう。

まず、メインループで、「時間設定ボタン」「スタートストップボタン」「カウントダウン終了」の3つのイベントを全て解析しています。そして、各イベントがヒットした際の処理は以下の通りです。

(1)「時間設定ボタン」イベントがヒットした際に、「時間設定状態」の場合は「カウントダウン時間設定」処理を行います。

(2)「スタートストップボタン」イベントがヒットした際に、「時間設定状態」の場合は「カウントダウン開始」処理を行い、「カウントダウン状態」に遷移します。「カウントダウン状態」の場合は「カウントダウン停止」処理を行い、「時間設定状態」へ遷移します。また「アラーム音出力状態」(その他)の場合は「アラーム音停止」処理を行い、「時間設定状態」へ遷移します。

(3)「カウントダウン終了」イベントがヒットした際に、「カウントダウン状態」の場合は「カウントダウン停止&アラーム音出力」処理を行い、「時間設定状態」へ遷移します。

状態については、マイナーループで表現するのではなく、「状態変数」により切り替えます。状態変数を用意しておき、遷移する際は、状態番号(時間設定状態(0)、カウントダウン状態(1)、アラーム出力状態(2))を設定します。そして、現在の状態を参照する際は、状態変数の内容を参照します。つまり、複数のマイナーループで状態を表現するのではなく、1つのメインループで表現し、状態を状態変数で切り替える構造になります。状態変数の切り替えにより、状態を遷移させるため、順序性のない“下部から上部へ状態を切り替える処理”なども容易に実装できます。

図8 キッチンタイマーをイベントドリブン型(E型)で実装

状態遷移表からイベントドリブン型(E型)で実装した場合と、基本仕様から直接実装した場合の比較



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!



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



スマートファクトリー化でCC-Link IE TSNが果たすべき役割

» Special 一覧

Special Site

- PR -



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



【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



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

https://monoist.atmarkit.co.jp/mn/articles/1211/07/news003_2.html

2/5

それでは、状態遷移表を作成し、その後、イベントドリブン型で実装した場合と、基本仕様から直接実装した場合を比較してみましょう。

状態遷移表からイベントドリブン型で実装した場合の特徴は、以下の通りです。

- ・ “マイナーループ” が存在せず、1つのメインループ構造になっている
- ・ 「スタートストップボタン」など、全てのイベントを1箇所で参照している
- ・ 状態変数の切り替えにより状態遷移を行っているため、順序性のない状態遷移にも適している

状態遷移表からイベントドリブン型で実装した場合、イベントが全て1箇所で管理されるため、イベントの解析モレはなくなり、品質が向上します。また、イベントを追加するなどの変更の際も、1つのメインループにイベントを追加すればよいため、メンテナンス性も向上します。そして、状態遷移表から実装したフローチャートは、もともと状態を意識した構造となっており、下部から上部へ移動するなど順序性のない状態遷移部分についても容易に実装することが可能です。

以上の特徴により、基本仕様書から直接実装した場合に挙げた3つの検討課題を解決できます。

- ・ **【検討課題1】: スタートストップボタンを1箇所で参照したい**
→ 解決
- ・ **【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**
→ 解決
- ・ **【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**
→ 解決

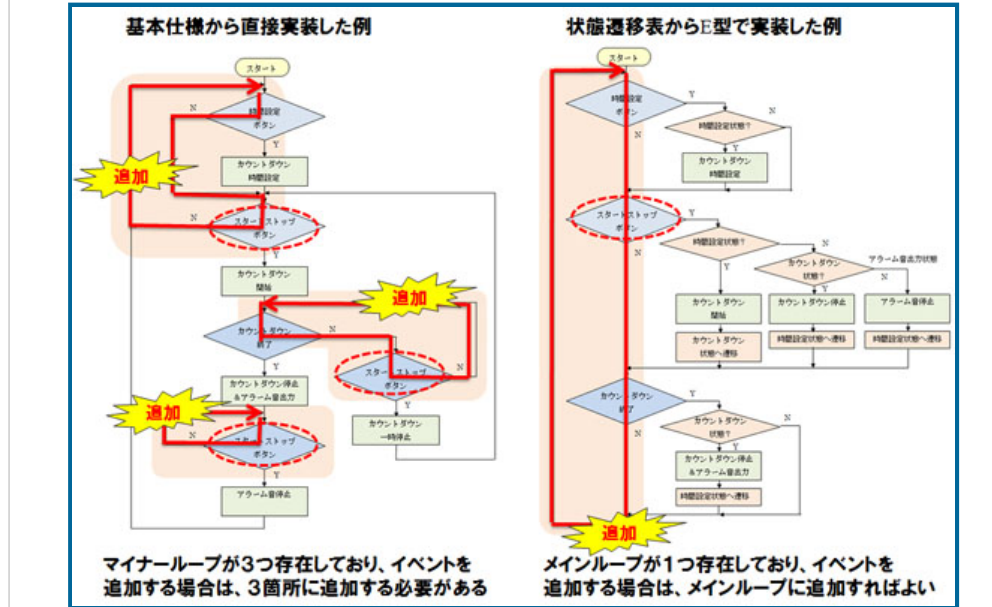


図9 状態遷移表からイベントドリブン型 (E型) で実装した場合と基本仕様から直接実装した場合の比較


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 ステートドリブン型 (S型)

<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

MONOist 主催セミナー

製造業×品質、転換期を迎えるモノづくりの在り方

開催日: 2019/5/29 (水) 会場: 野村コンファレンスプラザ日本橋

無料

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(3/3)

2012年11月07日 10時00分 公開

[塚田 雄一 キャッツ, MONOist]

前のページへ 1 2 3

印刷

通知

3

Share

7

ステートドリブン型(S型)

ステートドリブン型は、先に状態を解析するため、プログラムの1箇所ですべての状態を解析します。そして、状態を解析した後に、状態ごとに必要なイベントのみを解析します。例えば、アクションセルが“無視”や“不可”となるイベントのアクションについては解析を行いません。

主に、メニュー画面などのある画像処理分野では一般的な構造であり、多く使用されています。例えば、メニューA表示状態で、イベントXが入力された場合と、メニューB表示状態でイベントXが入力された場合などは、表示しているメニュー画面(状態)が確定していないと、イベントの意味が明確となりません。従って、最初に現在表示されているメニュー画面などの状態を解析した後に、メニューに応じた入力イベントを解析することが多く、そのようなシステムにはステートドリブン型の実装が適しています。

以下に、ステートドリブン型の特徴を記します。

- 常に全ての状態を解析する
- 必要なイベントのみを解析する
- アクションセルが“無視”や“不可”となるアクションセルのイベントの解析は行わない

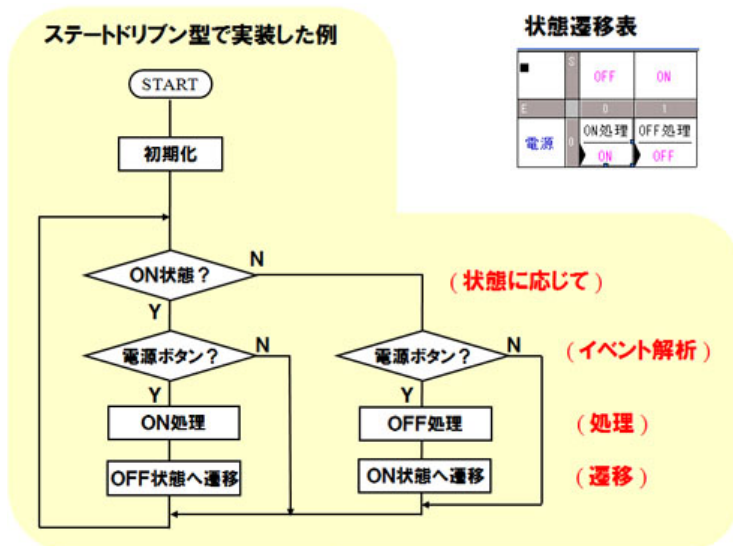


図10 ステートドリブン型(S型)で実装した例

カスタム検索



MONOist 主催セミナー

製造業×品質、転換期を迎えるモノづくりの在り方

開催日: 2019/5/29 (水) 会場: 野村コンファレンスプラザ日本橋

無料

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 -



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



コネクタ、センサーが実現する次世代モビリティ社会、「ホロレンス」で体験



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



スマートファクトリー化でCC-Link IE TSNが果たすべき役割



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!**



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

キッチンタイマーをステートドリブン型 (S型) で実装

それでは、状態遷移表からステートドリブン型で実装したフローチャートを作成してみましょう。

まずメインループで、「時間設定状態」「カウントダウン状態」「アラーム音出力状態」の3つの状態を全て解析します。この時、有効となる状態は1つのみです。つまり、メインループで状態の切り替えを行い、状態が遷移した際に、有効となるループを切り替える構造になります。そして、以下のよう

- (1)「時間設定状態」のループにおいて、「時間設定ボタン」イベントがヒットした際には「カウントダウン時間設定」処理を行い、「スタートストップボタン」イベントがヒットした際には「カウントダウン開始」処理を行い、「カウントダウン状態」へ遷移します。
- (2)「カウントダウン状態」のループにおいて、「スタートストップボタン」イベントがヒットした際には「カウントダウン停止」処理を行い、「時間設定状態」へ遷移します。「カウントダウン終了」イベントがヒットした際には「カウントダウン停止&アラーム音出力」処理を行い、「アラーム音出力状態」へ遷移します。
- (3)「アラーム音出力状態」のループにおいて、「スタートストップボタン」イベントがヒットした際には「アラーム音停止」処理を行い、「時間設定状態」へ遷移します。

状態については、イベントドリブン型と同様に、状態変数を用意しておき、遷移する際は状態番号 (時間設定状態 (0)、カウントダウン状態 (1)、アラーム出力状態 (2)) を設定します。そして、現在の状態を参照する際は、状態変数の内容を参照します。つまり、複数のマイナーループで状態を表現するのではなく、1つのメインループで表現し、状態を状態変数で切り替える構造になります。

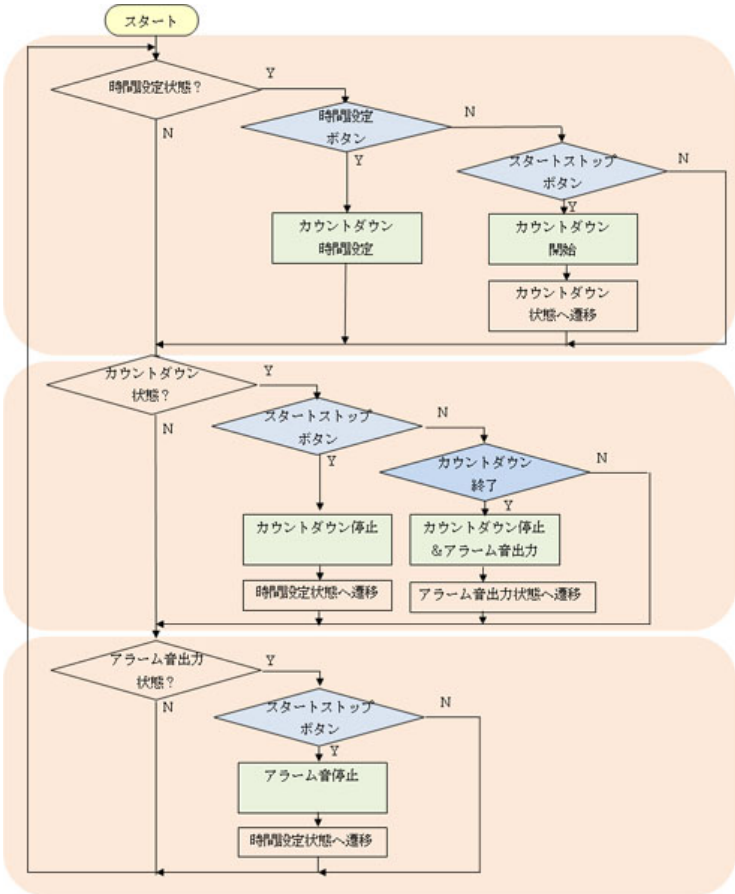


図 11 キッチンタイマーをステートドリブン型 (S型) で実装



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




自動車業界の大変革CASEに向け
て、マイクロソフトが持つ“部品”
New!


» Special 一覧

Special Site

- PR -




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




[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

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

状態遷移表からステートドリブン型 (S型) で実装した場合と、基本仕様から直接実装した場合の比較

それでは、状態遷移表を作成し、その後、ステートドリブン型で実装した場合と、基本仕様から直接実装した場合を比較してみましょう。

状態遷移表からステートドリブン型で実装した場合の特徴は、以下の通りです。

- ・ **“マイナーループ”が存在せず、1つのメインループ構造になっている**
- ・ **全ての状態を1箇所で参照している**
- ・ **状態変数の切り替えにより状態遷移を行っているため、順序性のない状態遷移にも適している**

状態遷移表からステートドリブン型で実装した場合、状態は全て1箇所で管理されるため、状態の解析モレはなくなり、品質が向上します。また、状態を追加するなどの変更の際も、1つのメインループに状態を追加すればよいため、メンテナンス性も向上します。そして、状態遷移表から実装したフローチャートは、もともと状態を意識した構造となっており、下部から上部へ移動するなど順序性のない状態遷移部分についても容易に実装することが可能です。

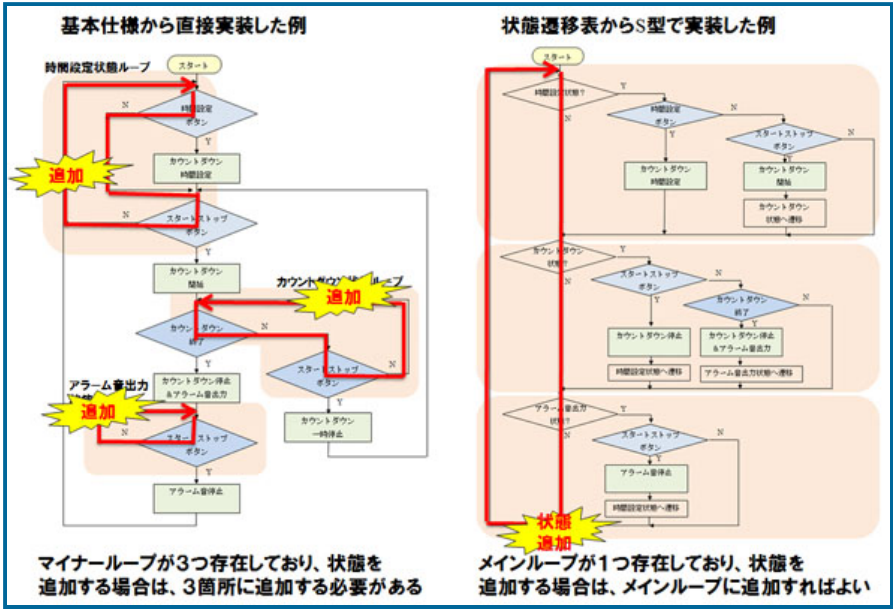


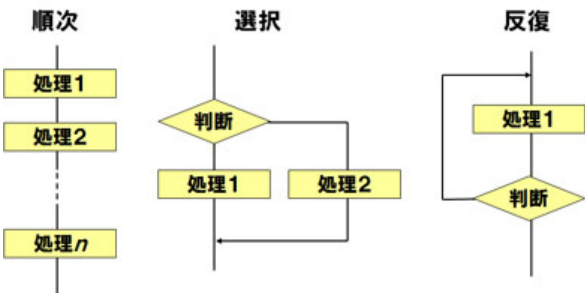
図12 状態遷移表からステートドリブン型 (S型) で実装した場合と基本仕様から直接実装した場合の比較

コラム: 構造化プログラミングとは

構造化プログラミングは、階層的に抽象化されたプログラムの組み合わせとして、プログラムを記述する手法であり、「順次」「選択」「反復」の3種類の制御構造のみを用いてプログラミングを行います。

構造化プログラミングの目的は、品質の良いプログラミング構造を実現することであり、無条件に、指定されたラベルや行番号にジャンプ (跳ぶ) するGOTO文を使用しません。GOTO文を使用すると、動作するシーケンスが複雑になるため、動作品質の保証が困難となります。

基本仕様から直接実装した場合のフローチャートにおいて、下部のループから上部のループにジャンプ (跳ぶ) するGOTO文を使用すれば、容易に実装できますが、上記の理由から、構造化プログラミングではGOTO文を使用せずに、フラグや変数でループを切り替えます。



- ☐ HPCとAI性能を両立したポスト「京」のCPU、ウエハーが初公開
- ☐ CAN通信におけるデータ送信の仕組みとは？
- ☐ 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
- ☐ 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
- ☐ 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

図13 構造化プログラミングとは

今回のまとめ

以下に、「基本仕様から直接実装」「状態遷移表からイベントドリブン型で実装」「状態遷移表からステートドリブン型で実装」を比較した表を記します(表1)。

比較項目	基本仕様から直接実装	状態遷移表からE型実装	状態遷移表からS型実装
ループ構造	マイナーループ3つを実行順序で管理	メインループ1つで管理	状態ごとのループをメインループで管理
状態遷移構造	上位ループから下位ループへ切り替え	状態変数で切り替え	状態変数で切り替え
イベント参照部分数	5箇所	3箇所	5箇所
イベント変更/追加	検討する場所がさまざま で困難	1箇所検討すればよい	3箇所検討する必要がある
状態変更/追加	検討する場所がさまざま で困難	3箇所検討する必要がある	1箇所検討すればよい
順序性のない状態への遷移	あまり向いていない	可能 ※容易にできる	可能 ※容易にできる

表1 「基本仕様から直接実装」「状態遷移表からイベントドリブン型で実装」「状態遷移表からステートドリブン型で実装」を比較

「基本仕様から直接実装」を行った場合は、イベント参照、状態参照など同じ処理があちこちに点在しています。それに比べ、状態遷移表は、イベントと状態を整理して考えられ、その後実装を行えば、イベントや状態などの処理をまとめて実装することが可能となります。また、状態遷移表から実装を行う手法として、イベントドリブン型とステートドリブン型が存在します。システムの性質上イベントによって処理を変えることが多く、イベント主体の処理であれば、イベントドリブン型で実装を行い、状態によって処理を変えることが多く、状態主体の処理であれば、ステートドリブン型で実装を行う方が適しています。

□

さて、今回は“状態遷移表からの実装”について解説しました。状態遷移表を用いると「モレ」「ヌケ」のない設計が行えるため、要求分析のフェーズで、検討すべき仕様が明確となります。また、状態とイベントの組み合わせが整理された状態遷移表(設計書)を作成し、それを基に、イベントドリブン型あるいはステートドリブン型で実装を行うことにより、変更が容易なソフトウェアを実現できるため、メンテナンス性が向上します。今回の内容から、状態遷移表設計手法が、設計レベルだけではなく、実装レベルにおいても、品質を向上させるために有効な設計手法であることが理解できたのではないのでしょうか。

さて、次回は「状態遷移表を使用したテスト手法」をテーマに、状態遷移表から試験項目、試験シナリオを作成する手法を紹介したいと思います。お楽しみに！(次回に続く)

組み込みモデリング コーナー



>>コーナーTOPはこちらから

「状態遷移表による設計手法」バックナンバー

状態遷移表を使用したテスト手法【後編】
状態遷移表を使用したテスト手法【前編】
状態遷移表からの実装
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)
状態遷移表を使用した要求分析モデル
なぜ状態遷移表を使うと、品質の良い開発ができるのか
状態遷移表設計手法の概要

この連載を「連載記事アラート」に登録する


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

関連記事

連載記事「状態遷移表による設計手法」

>>「組み込みモデリング」コーナー



「品質100%保証」を実現するCASEツールへ

「組み込みマルチコア進化論」最新記事一覧

前のページへ 1 2 3

組み込み開発

MONOist トップ

Copyright © ITmedia, Inc. All Rights Reserved.

Special Contents

- PR -



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



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



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



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



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!**



スマートファクトリー化でCC-Link IE TSNが果たすべき役割



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!**



「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

