

家電製品向けソフトウェア構造モデルの開発(2)

- シーケンス制御方式 -

5K-6

伊藤 謙次 内山 亘 岩本 恵子 関口 卓也 安武 剛一 田中 裕彦
松下電器産業株式会社

1 はじめに

我々は、家電製品の制御用マイコンのためのソフトウェア(以下、家電ソフトウェアとよぶ)の再利用性向上のために、家電ソフトウェア設計モデルとして機能関係モデルと、機能関係モデルにしたがった家電ソフトウェアの枠組として家電製品向けソフトウェア構造モデル [1][2] を提案している。

本稿では、家電ソフトウェア構造モデルの構成要素であるシーケンス制御部でのモジュールごとの独立性を向上させるシーケンス制御方式を提案する。

2 課題

処理行程の一連の流れ(以下、制御行程とよぶ)を一つの機能の単位として記述できれば制御行程の入れ換えにより、再利用性を向上させることができる。しかし、家電ソフトウェアではシーケンス制御の実行中であっても異常の発生やユーザの途中操作に対応するためにシーケンス制御を中断して行わなければならない処理(以下、例外処理とよぶ)が存在する。これは本来のシーケンス(以下、通常処理とよぶ)とは異なる流れの処理なので、通常処理の中断される可能性のある全ての箇所埋め込まなければならない。例えば、通常処理の処理1、処理2の直前で一時停止ボタンが押される可能性がある場合、図1に示すように、判定と中断の処理を埋め込まなければならない。

また例外処理の中では複数ハードウェアの一時停止や再開などを行う必要があるが、ハードウェアによっては他のハードウェアと関連しているため、ディレイを設けるなど一定の手順に従って処理を進める必要がある。

以上のように通常処理の途中で例外処理が埋め込まれているためシーケンス制御部のモジュール間の関係が複雑になる。

3 シーケンス制御方式

3.1 シーケンス制御方式の要件

シーケンス制御部の通常処理と例外処理を独立させるためのシーケンス制御方式の要件は以下のようになる。

A Software Model for Home Appliances(2) - Sequence Control - Kenji ITO, Wataru UCHIYAMA, Keiko IWAMOTO, Takuya SEKIGUCHI, Kouichi YASUTAKE, Hirohiko TANAKA
Matsushita Electric Industrial Co., Ltd.

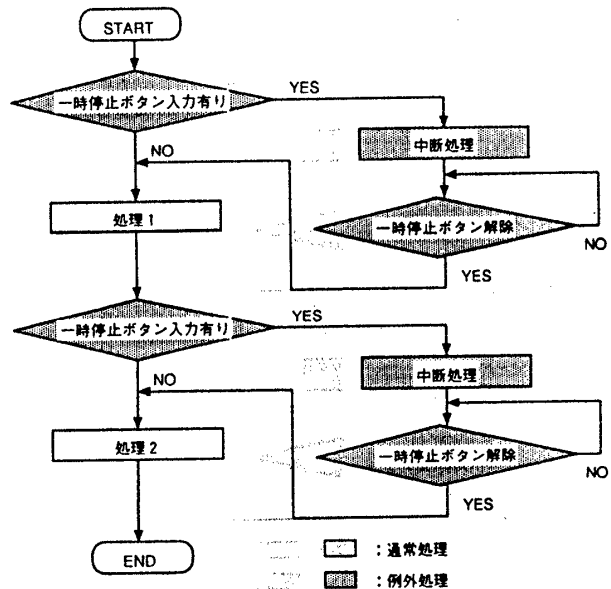


図1: 例外処理の記述例

1. 例外処理の発生を通常処理とは独立して検知できること
2. 例外処理の発生検知後、すみやかに例外処理を開始できること
3. 例外処理中は通常処理の実行を中断し、例外処理終了後には再開できること

例外処理終了後の通常処理の再開箇所によって例外処理を分類すると以下の3つに大別される。

- 一時停止型
例外処理が終了すると通常処理を中断した箇所から再開する。(例: 一時停止ボタン処理)
- 後戻り型
例外処理が終了すると通常処理を中断する以前の定められた箇所から実行する。(例: 電源ボタン処理)
- 停止型
例外処理中で実行を終了する。(例: 機器異常処理)

一時停止型では中断箇所のコンテキストを保存、復元する必要がある。後戻り型では再開すべき箇所のコンテキストを保存、復元する必要がある。停止型ではコンテキストを保存、復元する必要はない。

3.2 例外処理機構

家電組み込みマイコンはメモリ容量が小さいため、例外処理の発生の判定などの処理を各例外処理ごとにそのまま実装することは困難である。そこで、例外処理に必要な最小限の情報をテーブル形式で格納し、スケジューラがこのテーブルをもとに例外処理を実行させる。上記の要件を満たすには、例外処理ごとに以下の情報が必要である。

1. 例外処理の発生条件
2. 実行すべき例外処理プログラムの格納箇所
3. 例外処理終了後の通常処理の再開箇所
4. 通常処理再開時のスタック情報

1. についてはイベントフラグの真偽値を条件とすることで、メモリの低減をはかる。これにより例外処理の実行条件の判定期間の開始と終了を、例外イベント要求フラグの操作だけで指定することが可能となる。

マイコンによっては実装上の制約としてROMサイズよりもRAMサイズの方が厳しいこともある。一時停止型の場合、3.と4.の情報はプログラム実行後に動的に決定されるためこのテーブルはRAMに配置されなければならない。後戻り型の場合、3.と4.の情報はプログラム実行前に決定することが可能なので、ROMに配置することができる。停止型の場合、3.と4.の情報が不要である。

また、例外処理発生の判定は回路制御部からシーケンス制御部にタスク切替を行なう直前に行ない、判定の直後に例外処理を開始させることにより、例外処理の発生から回路制御部の割り込みの1周期以内に例外処理の実行を開始できる。通常処理の場合、回路制御部と通常処理用のシーケンス制御部の2つのタスクが実行されているが、例外処理中は、通常処理を実行する代わりに例外処理を実行させることにより、通常処理の中断が実現可能となる。

以上により、図2に示すように通常処理に例外処理を埋め込むことなく記述できるようになる。また、全ての例外処理を通常処理に埋め込む場合に比べROMサイズ、RAMサイズの低減が可能となる。

4 おわりに

家電ソフトウェアのシーケンス制御部を通常処理と例外処理とに分離するシーケンス制御方式と例外処理機構について報告した。

今後の課題としては、多重例外に対する処理の効率化などがあげられる。

参考文献

- [1] 関口卓也, 内山亘, 岩本恵子, 伊藤謙次, 縄田毅史, 安武剛一. 家電製品向けソフトウェア構造モデルの開発 (1)-スケジューリング方式-. 情報処理学会第50回全国大会, 5K-5, March 1995

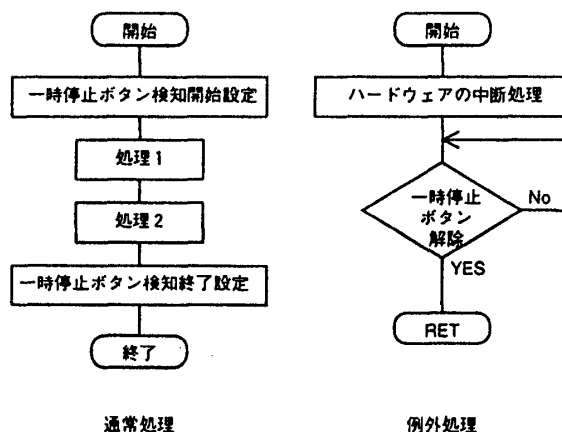


図2: 通常処理と分離した例外処理の記述例

- [2] 岩本恵子, 内山亘, 関口卓也, 伊藤謙次, 安武剛一, 黒木洋一. 家電製品向けソフトウェア構造モデルの開発 (2)-回路制御方式-. 情報処理学会第50回全国大会, 5K-7, March 1995