

# シーケンス・フロー図による組み込みソフトウェア設計の教育

舘 伸幸<sup>†1</sup> 香山瑞恵<sup>†2</sup>

**概要:** 組み込みシステムにおける制御ソフトウェア（以下、組み込みソフトウェア）では、リアルタイム処理や複数のタスクを同時実行させるなど、時間制約に対応することが求められる。これらは、リアルタイム OS（以下、ROS）を利用することで実現できる。しかし、その実現原理を理解しておくことは、組み込みソフトウェア技術者にとって重要である。今回我々は、特別な表記法を用いることで、順次実行の処理フローからリアルタイム処理と同時実行を実現するためのプログラム構造を導き出し、実際のプログラムを動作させて理解へ導く教材を作成し、情報系大学3年生62名に試行した。その結果、8割以上の学生が表記法を理解し、時間制約のあるプログラムを作成し、改造することができるようになった。

**キーワード:** 組み込みソフトウェア, リアルタイム, 同時実行, 教材

## Education of embedded software design by sequence flow diagram

NOBUYUKI TACHI<sup>†1</sup> MIZUE KAYAMA<sup>†2</sup>

### Abstract:

In control software in embedded systems (hereinafter referred to as embedded software), it is required to deal with time constraints such as real-time processing and simultaneous execution of multiple tasks. These can be realized by using real-time OS (hereinafter referred to as ROS). However, it is important for embedded software engineers to understand its realization principle, such as when ROS cannot be used or troubleshooting. In this time, we use a special notation method to derive a program structure for realizing real-time processing and simultaneous execution from the processing flow of sequential execution, create a teaching material that operates an actual program to lead to understanding, I tried 62 university third graders. As a result, more than 80% of students understood the notation, could create programs with time constraints, and be able to remodel.

**Keywords:** embedded software, real time, plurality of tasks, teaching material

## 1. はじめに

組み込みソフトウェアがそれ以外のソフトウェアと大きく異なる性質のひとつに、時間制約がある。このことは、組み込みソフトウェア技術の中でも、特に生産性に大きく影響を与えることが知られている[1]。すなわち、組み込みソフトウェア技術者にとって、時間制約に対応する技術は重要度が高いと言える。

我々は、enpit[2]と称する情報技術系大学生の教育事業に取り組んでおり、その中で組み込み技術分野の教育を実施している[3]。今回、9校62名の参加を得て合宿講義を行い、その中の学習テーマのひとつとして時間制約を取り上げた。本稿では、使用した教材の内容と実施結果について報告する。

## 2. 教材設計

### 2.1 課題と動作環境

課題は、2個のLED（以下、LED0とLED1）を、それぞれ異なる点滅パターンで動作させるものとした。LED0は単純に1秒周期で点滅するが、LED1は図1のように特殊

なパターンで点滅する。これらは、点滅の時間的正確さがリアルタイム処理に、異なるパターンでの点滅が同時実行に対応している。



図1 LED1の点滅パターン

Fig 1 Blinking pattern of LED 1

動作環境はArduino UNOを使い、LEDはそれぞれブレッドボード上に実装した。回路は実体配線図や配線した実物を示して準備しやすくした。

### 2.2 学習体制

学生は5名～6名を1チームとし、チーム内で相談しあえるようにした。座席は対面とし、チームに1面ずつ検討および発表用のホワイトボードを置いた。

### 2.3 行動目標

次のように行動目標を定めた。

- 時間を考慮した簡単な設計図が書ける
- 課題のプログラムを作り、改造できる

## 3. 教材の詳細と指導方略

時間制約を考慮した設計表記に、UMLではシーケンス図

<sup>†1</sup> 名古屋大学, 信州大学  
Nagoya University, Shinshu University  
<sup>†2</sup> 信州大学  
Shinshu University

やタイミング図が用意されている。詳細な時間関係を表記するには後者が向いているが、そもそも関係し合う処理単位（以下、部品）を分けることを理解している必要がある。そこで今回は、まず通常のフロー図で導入し、次に部品に分ける必然性ととも、時間制約を実現する方法を学ばせる方略をとった。

図 2(a)は LED0 と LED1 それぞれの点滅フローである。これらを同時に動作させるために、2 つのフローを合体させたものが図 2(b)である。この複雑な表記を経て、部品に分割した動作として、フロー間をメッセージでつないだ図 3 を説明することで、学生は部品分割の必然性と同時に、そのしくみで時間制約に対応できることを学ぶ。最終的に図 4 のようにメッセージ（の受け口）に命名することで関数として実装する工程に無理なく関連づけている。我々はこの表記方法を、シーケンス・フロー図と呼称している。

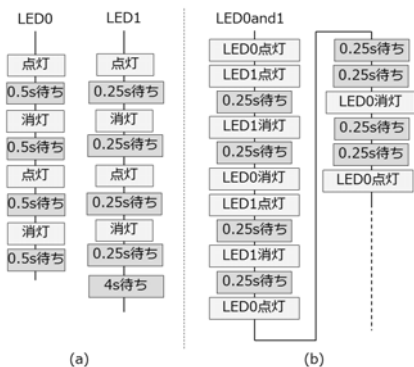


図 2 フローによる表現

Fig 2 Representation by flow diagram

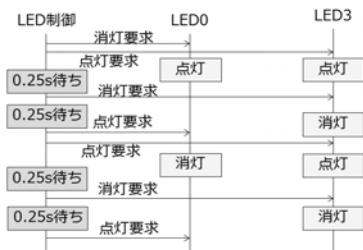


図 3 部品分割したフロー表現

Fig 3 Partial divided flow representation

授業では、まず LED0 の動作を例示し、その後 LED1 の設計を行わせ、実装させて動作を確認させた。また、動作の仕様変更を指示し、シーケンス・フローで改造方法を考えることで、容易に動作を変更できることを体験させた。

#### 4. おわりに

演習後のアンケートで、集中して聞いた（取り組めた）かについては、取り組めた 60.5%、まあまあ取り組めた 33.1%で、合計 93.6%の学生に課題について興味を持たせることができていた。

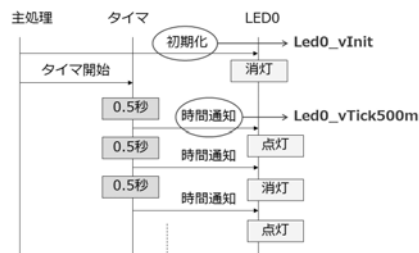


図 4 実装と関連付ける

Fig 4 Associate with implementation

また理解度については、理解できた 29%、まあまあ理解できた 57.3%で合計 86.3%が肯定的回答をしており、難易度の設定も適切であったと考えられる。

行動目標に対しては、「3.よくできる」から「1.あまりできない」まで3段階でアンケート調査した。簡単なシーケンス・フロー図が書けるようになったかについては、3 が 32.1%、2 が 53.6%で、合計 85.7%が肯定的回答をした。また、課題プログラムの作成と改造については3 が 34.5%、2 が 47.3%で、合計 81.8%が肯定的回答だった。

以上の結果から、時間制約への対応技術に関して、本教材を使った教育は一定の効果があると推定できる。

**謝辞** 今回教材実践の機会および計測結果のご提供をいただきました東海大学の渡辺晴美教授、佐藤未来子先生他、enPiT 組込み分野関連教員の皆様に感謝いたします。

#### 参考文献

- [1] 組込みソフトウェア開発データ白書 2017, 独立行政法人情報処理開発推進機構, 2017.11.14, <https://www.ipa.go.jp/files/000062317.pdf>
- [2] 平成 28 年度 大学教育再生戦略推進費 成長分野を支える情報技術人材の育成拠点の形成 (enPiT) 公募要領 [http://www.mext.go.jp/component/a\\_menu/education/detail/\\_icsFiles/fieldfile/2016/04/06/1369346\\_1\\_1.pdf](http://www.mext.go.jp/component/a_menu/education/detail/_icsFiles/fieldfile/2016/04/06/1369346_1_1.pdf)(参照 2018-05-01).
- [3] 成長分野を支える情報技術人材の育成拠点の形成組込みシステム分野 <http://emb.enpit.jp/enpit2/> (参照 2018-05-01)