



# *OpenSESSAME Seminar*

組込みソフトウェア技術者・管理者向けセミナー

初級者向けテキスト

組込みソフトウェア管理者・技術者育成研究会  
- SESSAME -

(<http://www.sesame.jp/>)

\*\*\*\*\* 目 次 \*\*\*\*\*

1 . SESSAME の紹介およびコースの概要	1
2 . 開発課題と失敗事例の解説	4
3 . 組込み向け構造化分析の例・設計の概要(1)	21
4 . 組込み向け構造化分析の例・設計の概要(2) 実習/回答と補足説明	44
5 . 組込み向け構造化設計(1)	51
6 . 組込み向け構造化設計(2) 実習/回答と補足説明	70
7 . プログラミング 組込み用語基礎知識	73
8 . ソフトウェアテストの概要	109
9 . プログラミング実習への説明	139
10 . プログラミング 実習	149
11 . プログラミング 実習の回答と補足説明	161
12 . ソフトウェアテスト 実習	164
13 . ソフトウェアテスト 実習/回答と補足説明	166
付録 . 話題沸騰ポットのシミュレーション	174

\*\*\*\*\*

## ソフトウェアテスト実習/回答と補足説明

西 康晴



1. SESSAMEの紹介およびコースの概要
2. 開発課題と失敗事例の解説
3. 組込み向け構造化分析の例・設計の概要(1)
4. 組込み向け構造化分析の例・設計の概要(2) 実習/回答と補足説明
5. 組込み向け構造化設計(1)
6. 組込み向け構造化設計(2) 実習/回答と補足説明
7. プログラミング 組込み用語基礎知識
8. ソフトウェアテストの概要
9. プログラミング実習への説明
10. プログラミング 実習
11. プログラミング 実習の回答と補足説明
12. ソフトウェアテスト 実習
13. ソフトウェアテスト 実習/回答と補足説明

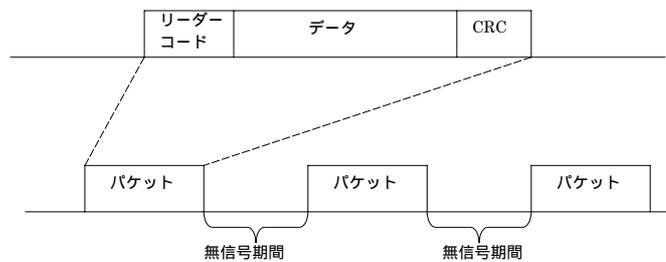
付録：話題沸騰ボットのシミュレーション

331

## 通信データフォーマット

### ・ パケットと無信号期間がある

- リーダーコードは不明
- データは 4バイト
- CRCは 1バイト
- 無信号期間は不明



332

## 境界値テストの設計

- データ部の境界値テスト
  - 4バイトのデータ部を送る 正常
  - 4バイト以上のデータ部を送る はみ出てCRCエラーとなる
    - 実際にはバッファオーバーフローが起きるかもしれない
  - 4バイト以下のデータ部を送る 足りずにCRCエラーとなる
- CRC部の境界値テスト
  - 1バイトのCRC部を送る 正常
  - 1バイト以上のCRC部を送る 無視される
    - 実際にはバッファオーバーフローが起きるかもしれない
  - 1バイト以下のCRC部を送る CRCエラーとなる
- リーダーコード/無信号期間の境界値テスト
  - 境界値が分からない場合は問い合わせる
    - 分からない仕様はバグの温床

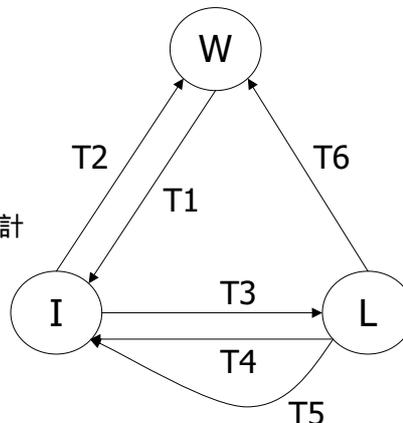


333

SESSAME CONTENTS 2004

## 状態遷移パステストの設計

- 遷移リンクの抽出
  - T1: W I
  - T2: I W
  - T3: I L
  - T4: L I (受信完了)
  - T5: I L (受信エラー)
  - T6: L W
- リンク網羅での状態遷移パスの設計
  - P1: T1 T2
  - P2: T1 T3 T4
  - P3: T1 T3 T5
  - P4: T1 T3 T6

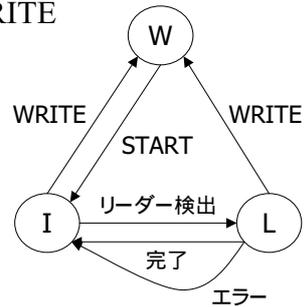


334

SESSAME CONTENTS 2004

## 状態遷移パステストの設計

- イベントに直してテスト項目を記述する
  - P1: START WRITE
  - P2: START リーダー検出 受信完了
  - P3: START リーダー検出 受信エラー
  - P4: START リーダー検出 WRITE
- 期待結果を記述する
  - P1: WRITE状態
  - P2: データを送出してIDLE状態
  - P3: IDLE状態
  - P4: WRITE状態



335

SESSAME CONTENTS 2004

## 状態遷移マトリクス

遷移  
/ 無視  
x ありえない

状態 イベント	WRITE (受信機のパラメータ 設定モード)	LOCK (データ受信モード)	IDLE (リーダーコード 待ちモード)
WRITE (SozeXから)	/	WRITE	WRITE
START (SozeXから)	IDLE	/	/
リーダー検出 (セルフイベント)	x	x	LOCK
受信完了 (セルフイベント)	x	IDLE	x
受信エラー (セルフイベント)	x	IDLE	x

336

SESSAME CONTENTS 2004

## 状態遷移マトリクステストの設計

- 各状態でのイベントをそれぞれテストする
  - 無視されるイベントも確認する
    - 発生しうるイベントは6つ
    - 無視されるイベントは3つ
  - ありえないイベントはデザインレビューで確認する
    - ありえないイベントは6つ
  - テストドキュメントを作成するに越したことはないが、経験が無いなら状態遷移マトリクスにOKを書く
    - NGはきちんと不具合報告書で報告すること
- イベントの排他発生を確認する
  - WRITE / START イベントは排他
  - リーダー検出 / 受信完了 / 受信エラー イベントは同時に発生する可能性がある
    - 同時に発生した時にフェイルセーフになるかどうかを確認する必要がある

OK結果の  
記録も重要



337

SESSAME CONTENTS 2004

## 状態遷移マトリクステストの設計

- 状態遷移マトリクスからテスト項目を設計する
  - 発生しうるイベントのテスト項目を設計する
    - WRITE状態でSTARTイベントを起こし、IDLE状態に遷移する
    - LOCK状態でWRITEイベントを起こし、WRITE状態に遷移する
    - LOCK状態で受信完了イベントを起こし、データを送出してIDLE状態に遷移する
    - LOCK状態で受信エラーイベントを起こし、IDLE状態に遷移する
    - IDLE状態でWRITEイベントを起こし、WRITE状態に遷移する
    - IDLE状態でリーダー検出イベントを起こし、LOCK状態に遷移する
  - 無視されるイベントのテスト項目を設計する
    - WRITE状態でWRITEイベントを起こし、無視されることを確認する
    - LOCK状態でSTARTイベントを起こし、無視されることを確認する
    - IDLE状態でSTARTイベントを起こし、無視されることを確認する
  - 同時に発生するイベントのテスト項目を設計する
    - 各状態でリーダー検出、受信完了、受信エラーイベントを起こし、受信エラー動作が起こる(受信完了動作が起こらない)ことを確認する



338

SESSAME CONTENTS 2004

## Irレシーバーサブシステムのソースコード

```

0      for (;;) {
1          /* determine event */
2          event = IR_EVENT_NUM;
3          hostCmd = GetHostCmd();
4          if (hostCmd == WRITE)
5              event = EVENT_WRITE;
6          else if (hostCmd == START)
7              event = EVENT_START;
8          else {
9              GetIrEvent(&leadDetectF, &dataReadyF, &receiveErrorF);
10             if (receiveErrorF)
11                 event = EVENT_RECEIVE_ERROR;
12             else if (leadDetectF)
13                 event = EVENT_LEAD_DETECT;
14             else if (dataReadyF)
15                 event = EVENT_DATA_READY;
16         }
17
18         if (event != IR_EVENT_NUM) {
19             callp = dataTbl[current_mode][event].fp;
20             next_mode = dataTbl[current_mode][event].mode;
21             (*callp)();
22             current_mode = next_mode;
23         }
24     }

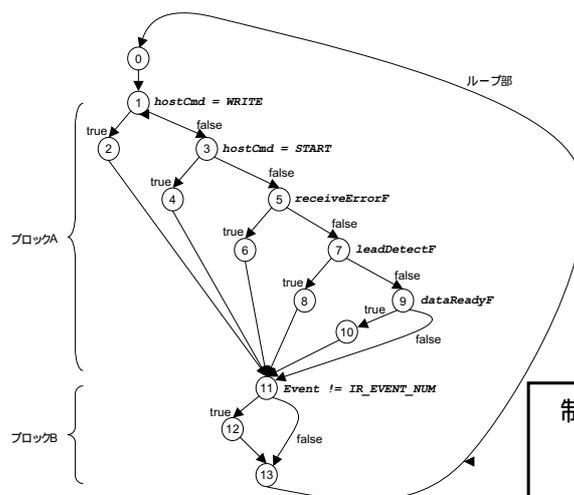
```



339

SESSAME CONTENTS 2004

## Irサブシステムの制御フローグラフ

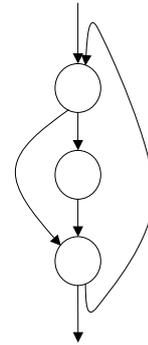


340

SESSAME CONTENTS 2004

## なぜリンク網羅が必要かを理解する

- ノード網羅は ○ しか網羅しないのでバグ見逃しが起こる
  - 9 11 / 11 13のような「飛び越し」をテストできない
  - ループもテストできない
- リンク網羅は → を網羅する
  - 飛び越しもループもテストできる
  - パスもテストできる
- ループをテストするときは (0回、)1回、2回、たくさん、のテストをする
  - 要はリンク網羅である

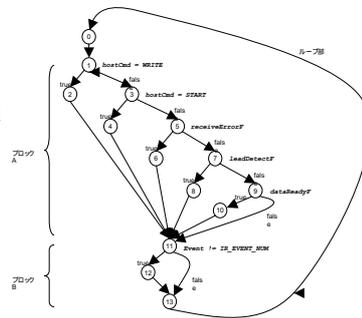


341

SESSAME CONTENTS 2004

## 3つの部分に分けてパスを抽出する

- ブロックAの部分パス
  - A1: 0 1 2 11
  - A2: 0 1 3 4 11
  - A3: 0 1 3 5 6 11
  - A4: 0 1 3 5 7 8 11
  - A5: 0 1 3 5 7 9 10 11
  - A6: 0 1 3 5 7 9 11
- ブロックBの部分パス
  - B1: 11 12 13
  - B2: 11 13
- ループ部の部分パス
  - L0: (ループ0回) 無し
  - L1: 0 1... 13 (1回)
  - L2: 0 1... 13 0 1...13 (2回)
  - L : 0 1... 13 0 1...13 ... 0 1...13 0 1...13



342

SESSAME CONTENTS 2004

## 組み合わせを検討する

- ブロックAとブロックBに依存関係があるか
  - A1～A5は必ずB1に、A6は必ずB2に分岐する
- ブロックA/Bとループ部に依存関係があるか
  - ブロックの履歴に動作が依存しないか
    - 過去のブロックA (B)とブロックB (A)
    - 過去の自分自身
  - 今回は依存しないと想定する
- デザインレビューやコードインスペクションを行う
  - 保守や拡張、後継機種開発の時にも気を付ける



343

SESSAME CONTENTS 2004

## テストパスの設計

- L1のパス:
  - L1-1: A1 B1
  - L1-2: A2 B1
  - L1-3: A3 B1
  - L1-4: A4 B1
  - L1-5: A5 B1
  - L1-6: A6 B2
- L2:
  - 履歴による依存関係はないので、L1-1～L1-6を適宜組み合わせればよい
    - 依存関係がある「疑い」がある場合は6つのパスを全て組み合わせなければならない

### 設計したテストパス

A1 B1 (WRITE)  
A2 B1 (START)  
A3 B1 (受信エラー)  
A4 B1 (リーダー検出)  
A5 B1 (受信完了)  
A6 B2 A1 B1  
(イベントなし WRITE)



344

SESSAME CONTENTS 2004

## テスト実施結果の確認方法

- ICE / デバッガでテストする場合
  - ループごとに実行を停止させて変数 event と current\_mode (next\_mode) をウォッチする
  - printf などアサーションコードを入れる
- モジュールの引数 / 返値でテストする場合
  - LOCK 状態で dataReadyF を発生させてテスト結果を確認する必要がある
    - こうなるとコードからテストケースを起こすのは至難の業となる
    - コードが設計と同じ構造をしていれば、状態遷移パスで考えることができ楽である
- モジュールをブラックボックスにしないように気を配る
  - テスト結果を観測しやすい設計や実装を意識する



345

SESSAME CONTENTS 2004

## ソフトウェアテスト実習のまとめ

- それぞれのテストの特徴を掴もう
  - 境界値テストでは、通信データのテストが設計できた
  - 状態遷移パステストでは、イベントの順序のテストが楽に設計できた
  - 状態遷移マトリクステストでは、同時イベントのテストが設計できた
  - 制御パステストでは「イベントなし」のテストが設計できた
- テストの設計と同時に、仕様レビュー / デザインレビュー / コードレビューを行っていることを肝に銘ずる
  - 期待結果の導出と、不明な期待結果の問い合わせを忘れない
  - テスト設計の時点で、仕様や設計、実装のバグを見つけよう
  - テストの組み合わせを減らす設計やレビューを心がける
  - テスト結果を観測しやすい設計や実装を意識しよう



テストが上手になってくると、  
そもそもバグのない開発が  
できるようになる

346

SESSAME CONTENTS 2004

## 本ドキュメントのご利用に際して

- 本著作物の著作権は作成者または作成者の所属する組織が所有し、著作権法によって保護されています
- SESSAME は本著作物に関して著作者から著作物の利用 を許諾されています
- 本著作物は SESSAME が利用者個人に対して使用許諾を与え、使用を認めています
- SESSAME から使用許諾を与えられた個人以外の方で本著作物を使用したい場合は [query@sessame.jp](mailto:query@sessame.jp) までお問い合わせください

SESSAME が著作者から許諾されている権利

著作物の複製・上演・演奏・公衆送信及び送信可能化・口述・展示・上映及び頒布・貸与・翻訳・翻案・二次的著作物の利用

- ドキュメント中には Microsoft 社, Adobe 社等が著作権を所有しているクリップアートが含まれています

---

### OpenSESSAME Seminar

#### 組込みソフトウェア技術者・管理者向けセミナー 初級者向けテキスト

---

2002 年 10 月 15 日 初版 第 1 刷発行

2003 年 10 月 29 日 初版 第 2 刷発行

2004 年 3 月 19 日 第 2 版 第 1 刷発行

2004 年 4 月 30 日 第 3 版 第 1 刷発行

2004 年 6 月 17 日 第 4 版 第 1 刷発行

**著 者** 上原慶子、大野晋、坂本直史、鈴木圭一、須田泉、西康晴、  
二上貴夫、三浦元、三宅貴章、森孝夫、山田大介、山崎辰雄

**編集・発行** 組込みソフトウェア管理者・技術者育成研究会  
(SESSAME)

<http://www.sessame.jp>

---

無断転載・複写、使用を禁ず

Printed in Japan