「SWP1-D301　ムーブレボ通報システム　ソフトウェア要求仕様書」（山本雅基版）の解説

2013 年 5 月 27 日

1．作成の経緯

ASDoQ 人材育成部会では，2012 年度に開発文書の事例作成に取り組みました．「SWP1-D301　ムーブレボ通報システム　ソフトウェア要求仕様書」(山本雅基版)は，その成果の一部です．

開発文書には，要求仕様書や設計書など複数の種類がありますが，人材育成部会では，2012 年度に「ソフトウェア要求仕様書」事例を作成することにしました．そして，ソフトウェアが搭載されるシステムは，近未来のパーソナル移動体の通報システムとしました（3 章参照）．

なお，近未来のパーソナル移動体は，ムーブレボと呼称します．

2．作成方法

(1)上位文書

組込みシステムでは，そのソフトウェア要求仕様書を作成するためには，システム要求仕様書やシステムアーキテクチャ設計書，さらにはハードウェア設計書などという上位文書や関連文書が必要になります．

しかし，今回は，それらの上位文書を作らずに，直接ソフトウェア要求仕様書の作成に取り組みました．そのために，上位文書や関連文書と書かれている内容を仮定し，ソフトウェア要求仕様書では，それらを参照する箇所を明記しました．

(2)個別作業

ソフトウェア要求仕様書は，部会メンバが個人作業で作成しました．

さらに，毎月に開催した人材育成部会議の場で，作成途中の内容に関して意見交換をしました．

(3)参考にした文書

以下の 2 文書を参考にしました．

- IPA/SEC「【改訂版】 組込みソフトウェア向け開発プロセスガイド」2007
  http://sec.ipa.go.jp/publish/tn07-005.html
- IEE830

(4)目次の構成

前記に挙げた参考文献には，ソフトウェア要求仕様書の目次構成が提案されています．そこで，当初は，それらの目次構成に準拠して作成しました．

しかし，完成した文書の目次構成は，筆者の今までの経験や好みを反映させ，そのいずれとも異なるものにしました．筆者なりに，ムーブレボ通報システムのソフトウェア要求を考え続けた結果です．なお，作成した文書は，参考文献に述べられた記述項目をほぼ満たしていることを確認していますが，その内容については議論の余地があります．

もちろん，前記の参考文献で提案された目次をより尊重して，その目次構成に近い目次

の事例を作成することも可能であると，考えます．どなたかが，そのような文書を作成することを，期待します．

3. ムーブレボ通報システム

想定したムーブレボ通報システムの概要を，以下に示します．

(1)位置づけ

ムーブレボ通報システムは，ムーブレボに搭載されており，異常発生時にムーブレボセンターへその旨を通報する装置です．

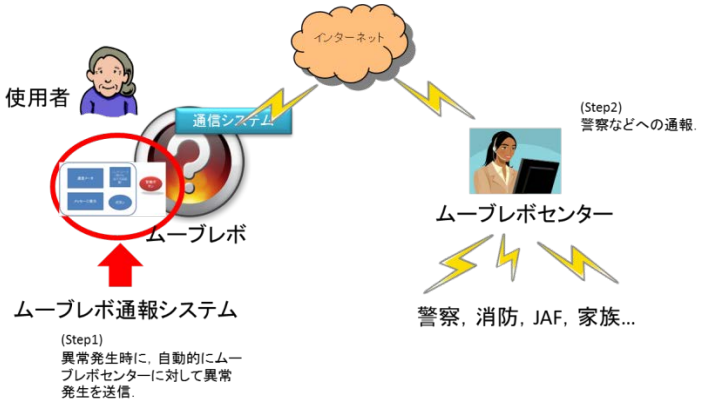

図 1　ムーブレボ通報システムの位置づけ

(2)通報システムの構成

通報システムは，G センサ，GPS センサ，通信システムなどとインタフェースを持ちます．図 2 に，通報システムの構成図を示します．

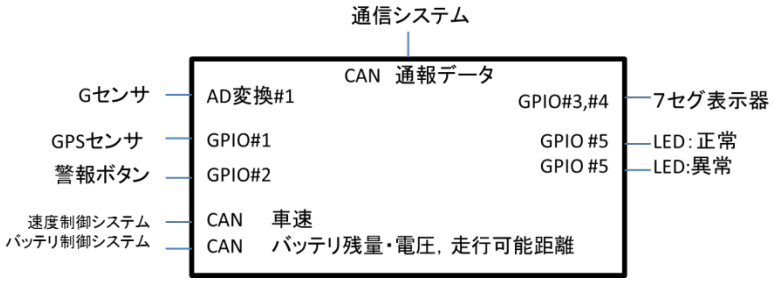

図 2　構成図

(3)通報システムの要求仕様

通報システムの要求仕様として，以下を想定することにしました．

- 通報タイミング
  - 正常時の通知
  - 5 秒間隔
- 異常時の通報
  - 乗員が警報ボタンを押下，または G センサが 3 G 以上の衝撃を検出

- 通報データ
  - <ムーブレボ管理番号>,<通報事由>,<発生日時>,<緯度経度座標>
    - 通報事由： 正常，過度の G，または警報ボタン押下
- Status 表示
  - 正常：正常時
  - 異常：乗員が警報ボタンを押下，または G センサが 3 G 以上の衝撃を検出
- リセット
  - コントローラ裏面の小さな穴の奥にあるリセットボタン押下により，ハードウェア的にリセットする．
- システムの品質
  - 略．
  - ソフトウェアに対しては，以下を要求する．
    - ASDoQ 社の標準開発プロセスに準拠して開発すること
    - ソフトウェアテストのコードカバレッジは，MC/DC とすること
    - 異常時は，検出から 0.2 秒以内に，通報すること

(4)通信データ

- GPS センサ I/F
  - シリアル通信で GPS 値などを獲得する．
  - 周期：1Hz
  - 電文：33 バイト：<1><2><3><4><5><6><CR><LF>
    - <1>:9 バイト： UTC 時刻 hhmmss.ss
    - <2>:9 バイト：緯度 ddmm.mmmm
    - <3>：1 バイト： N=北緯，S=南緯
    - <4>:10 バイト：経度 dddmm.mmmm
    - <5>：1 バイト：E=東経，W=西経
    - <6>：1 バイト：有効性 0=無効，1=有効
- G 値 I/F
  - G 値を電圧値として獲得する．AD 変換して，G 値を求める．
  - 電圧と G 値の関係
    - 0v = 0G .. 4v = 5G, この間は，線形補間
    - 最小単位： 0.05v
    - G センサ異常時は，5v の値を取る
- 警報ボタン I/F
  - ポートのビット値として読み込む
  - 取り得る値： ON(押下):1, OFF:0
  - 読み込み時の注意：値が変更した時点から 0.01 秒後に再度読み込み，同じ値の時に，ボタン状態が変更したとみなす．

- 速度制御機 I/F
  - 車速信号を CAN で取得する
  - 周期：100msec
  - ID： 0x080
  - データ長：1 バイト
  - データ： 0..100 , 1=1km/h (例: 0=0km/h, 20=20km/h)
- バッテリ制御機 I/F
  - バッテリ残量と走行可能距離を CAN で取得する
  - 周期：10sec
  - ID： 0x90
  - データ長：2 バイト
  - データ：第 1 バイト，残量 0..100 , LSB 1% (例: 0=0%, 20=20%)
    第 2 バイト，電圧 0..255, LSB 15/255v(例： 204=12v)
    第 3 バイト，走行可能距離 0..100 ,LSB 1km (例: 0=0km, 20=20km)
- 7 セグ表示器 I/F
  - GPIO#2 ：8 個（速度 2 個，残量 3 個，走行可能距離 3 個）の切り替え
  - GPIO#3 ：各セグメントの ON/OFF 制御
- 状態通知 LED I/F
  - ポートにビット値を書き込む
  - 取り得る値： LED_ON:1, LED_OFF:0
  - 対応するポート： GPIO#4 bit1:正常 LED，bit2:異常 LED
- 通信モジュール I/F
  - 入力
    - 通信状態を CAN で取得する
    - ID: 0x040
    - データ長：1 バイト
    - データ： 1=異常データの送信済み TBD
  - 出力
    - 正常時および異常時の現在位置などを CAN で出力する
    - ID: 0x008
    - データ長：7 バイト TBD
    - データ： 通報事由 1 バイト（正常時|警報ボタン | G 超過）
      緯度経度座標 6 バイト

以上