C-005

センサシステム開発のためのデバッグ支援ツールの提案
Suggestion of the debugging support tool for sensor system development

石原 一輝† 平山 雅之†
Kazuki Ishihara Masayuki Hirayama

1. はじめに

センサを用いたシステム開発時には，センサの出力を考慮したプログラムの作成が必要となる．我々はこれまでに組み込みシステム開発の題材として，自転車事故防止システムの開発を進めている[1]．自転車事故防止システムとは，自転車搭乗者の行動や，周辺の道路環境から危険行動や危険環境を自転車各部に設置したセンサによって検出し，警告を行うシステムである．システムによって事故につながる行動や環境を正しく認識するためには，各種センサの出力について詳細な分析を行う必要がある．センサの出力データの分析方法として，マイコンと PC 間で通信を行い，データを可視化する方法があるが，判定部の作成は，センサの出力値に依存しているため，プログラムの書き換えなどにより，時間と手間のかかる工程となっている．

本研究ではセンサシステム開発時に作成される計測・判定プログラムに着目し，マイコンと PC 間の通信時におけるセンサデータの表示方法や利用容易な開発方法について検討する．

2. 研究背景

2.1 組込みシステム開発における printf デバッグ

printf デバッグとは，プログラム内部に printf などの出力関数を挿入することで，出力された文字列からバグの原因を究明する手法である． 文字列の形式(“A” “B”)や有無(“A” “”)，順番(“AB” ”BA”)などから，プログラムの内部処理を理解するもので，どの位置にどの程度デバッグ文を挿入するかは，個人の裁量によるため慣れが必要となる．一方，プログラム内部に挿入された printf は，マイコン内部で他の命令と同様に処理されるため， printf の挿入や削除によってシステムの実行時間が変化したり，新たにコンパイルやマイコンへの書き込みが必要になるといった問題を抱えている．printf を用いないデバッグ手法として，各種組み込み CPU 向けのデバッガを利用する方法[2]があるが，printf デバッグは，利用が容易であることや，デバッグ結果が文字として残るといった特徴から洞察的なデバッグが可能であることから，依然として有効なデバッグ手法として用いられている．

2.2 自転車事故防止システムの開発

自転車事故防止システムは，センサを利用したシステムであり，センサの出力によって危険運転や危険環境の検出を行っている．検出部の開発では，センサの出力を用いたデバッグ・テスト作業が必要となり，PC や LCD キャラクタディスプレイといった表示デバイスを用いて，プログラムの実行結果を文字列として表示させることにより，その確認作業を行っている．その際，文字列の表示方法として以下のような方法が存在することが確認できた．

- フロー表示(TeraTerm[3]を用いた PC 表示)
  マイコンから送信された文字列が画面上の新たな行に次々に追加されていく
- 固定表示(LCD キャラクタディスプレイによる表示)
  画面を一定時間間隔で更新し，固定した位置で文字を表示させる

フロー表示はプログラムの処理の流れを把握するのに有効であるが，いくつもの変数を同時に確認するのが困難である．一方，固定表示は，センサなど頻繁に書き換わる変数を確認するのに有効であるが，表示位置を考慮したプログラムの設計が必要である．センサデータの数値処理を行う点から，これまで主に PC への表示を行っており，1 秒毎にセンサデータを表示させる場合など，表示させる文字列の量が多くなると画面が文字で埋め尽くされ，変数の表示される位置が変動するなど，センサの数値の確認が困難になる場面が見られている．

3. デバッグ支援ツールの概要

提案するデバッグ支援ツールは，printf によるデバッグ作業を支援するものであり，一般的なシリアル通信ソフトの機能に加えて，以下の 2 つの機能を有している．

(1) 文字列種による表示方法の変更
(2) センサの出力に同期したユーザー関数の実行

3.1 文字列種による表示方法の変更

過去の開発経験から，マイコンから出力関数を用いて PC 上へ送信される文字列は，その特徴から図 1 のように分類することができる．

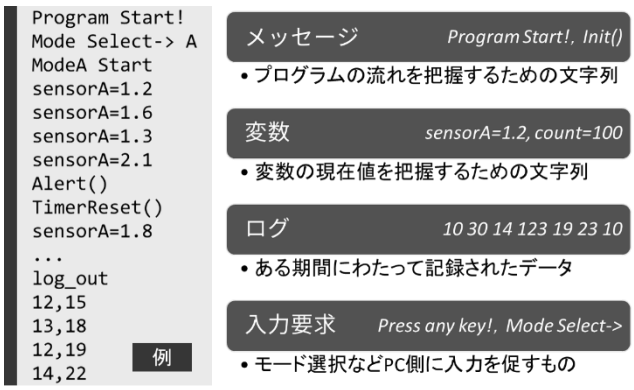

図 1 デバッグ文字列の種別

従来のシリアル通信ツールでは，これらの文字列が区別されることなく，図 1 のように一律にフロー形式によって表示が行われている．特に”変数”については，センサの出

†日本大学大学院理工学研究科
Graduate School of Science and Technology, Nihon University

力値を主な表示文字列としたとき，リアルタイムなセンサの出力値を把握するため，定期的にその出力が行われる．そのため，表示部の視認性から，処理の流れを表す”メッセージ”と，センサなどの出力値を表す”変数”を同時に出力されるといったことはあきらめ，片方のみを出力させるといった方法をとることがある．

提案ツールは，これらデバッグ文字列の種類をツール側で判定することによって，これら文字列を混在させた場合でも，デバッグの行いやすい表示方法を提供する．各文字列は，それぞれ出力順を把握する目的で従来通り，フロー形式によって表示を行うが，図 2 のように，追加された文字列のレイアウトはそれぞれ異なったものとなる．”変数”は，その値が頻繁に追加されることから表示非表示を切り替えることができ，固定した位置での表示も可能となっている．また，センサシステム開発時に行われるしきい値決め作業を支援する仕組みとして，しきい値の設定画面を設けており，出力値に応じて画面の色が変化するといった機能を備えている．プログラム内部のモードを遷移させる方法として，PC 側から入力を受け付けることがあるが，入力を受け付ける文字列を予め，”入力要求”によって指定することで，ボタンを複数配置した選択形式による表示が行われる．

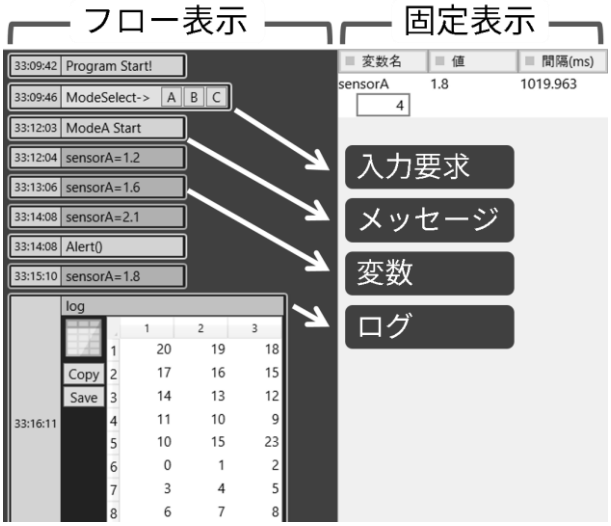

図 2　デバッグ文字列の表示画面

3.2 センサの出力に同期したユーザー関数の実行

センサを用いた判定プログラムの作成に焦点をあて，デバッグ作業の効率化を図る．センサを用いて，ある事象を検出しようとしたとき，センサの出力を計測するプログラムとセンサの出力から事象を判定するプログラムの 2 種類のプログラムが作成される．このとき判定プログラムの内部処理は，計測プログラムと比べて複雑になりやすく，プログラム内に記述ミスがあったり，判定のアルゴリズムに間違いが入りやすくなる．そういった場合，コードのコンパイル，マイコンへの書き込み，マイコンの再起動などが必要となり，該当部のデバッグ作業は，より一層煩雑なものとなる．提案ツールでは，そういったセンサを用いた判定プログラムを対象にしたコードシミュレータを提供する．

コードシミュレータでは，予めセンサを用いた判定プログラムの挙動を PC 上でシミュレーションすることにより，マイコンで実機動作させた際の不具合の発生を低減させることを目的としている．ツールによって”変数”は変数名と値のペアによって記録されているため，その変更と値を利用することで，ユーザーが作成した関数を呼び出すといったことが可能となる．

図 3 のように，まず画面中央のテキストエディタに適当な関数を記述し，その引数としてマイコンから送信された変数に対応する変数名を記述する．画面上部ではツールによって過去に取得された変数の一覧が確認可能であり，そこから変数名を選択することによって，ユーザー関数のひな型を自動生成することも可能である．その後，コードを実行させると，作成したユーザー関数の呼び出しが変数の変更のタイミングで行われる．コードの実行は瞬時におこなわれるため，実行結果をもとにコードの修正を容易に行うことが可能である．図の例では，変数名が「sensor0」である”変数”が 50 件受信されるごとに，ユーザー関数「function()」の呼び出しが行われており，受信された変数である「sensor0」の合計値と平均値の算出を行っている．

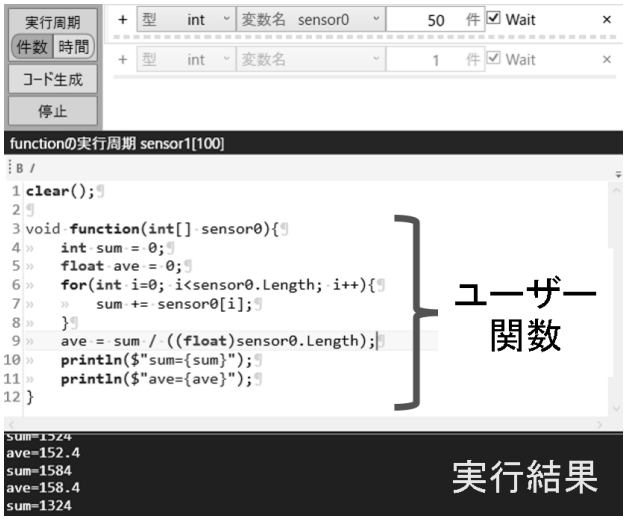

図 3　ユーザー関数の作成実行画面

4. まとめ

組込みシステム開発時に用いられるシリアル通信ツールによる printf デバッグを対象にしたデバッグ支援ツールの提案を行った．本ツールでは，プログラム内で記述するデバッグ文を 4 種類に分類することで，デバッグ時のデータの利用容易性を向上させるものとなっている．またそれに付随して，利用者が作成したコードを簡易的に実行できる仕組みをシリアル通信ツールの一部として実装を行った．今後は，提案ツールの評価実験を進めていく．

参考文献

[1] 山崎 和人, 宮澤 雄介, 平山 雅之, “センサを用いた走行パターン解析による自転車事故防止システムの提案”, 第 77 回全国大会講演論文集, (2015)
[2] RENESAS「デバッガ/エミュレータ」 http://japan.renesas.com/products/tools/emulation_debugging/
[3] ターミナルエミュレータ「TeraTerm」 https://ttssh2.osdn.jp/