

牛舎内酪農機器通信制御システムの開発

堤 大祐, 山本 寧, 川口 隆*
深谷 政廣*

Development of Network System for Controlling Dairy-machines in Barns

Daisuke TSUTSUMI , Yasushi YAMAMOTO , Takashi KAWAGUCHI*
Masahiro FUKAYA*

抄 録

牛舎内には自動給飼機、乳量計などの省力化を目的とした酪農機器があり、現在、それぞれは独立に動作している。酪農作業の省力化を図るため、各酪農機器に通信機能をもたせ、パソコンと酪農機器を通信ネットワークで接続して制御する通信制御システムを開発した。本通信制御システムは、母屋などにパソコンを置いて、酪農機器の遠隔操作・遠隔監視を可能にするものであり、搾乳量などのデータを収集したり、給飼量の設定を容易にするものである。給飼量は給飼時刻ごと、牛ごと、飼料の種類ごとに設定する必要があり、パソコンの画面からマウスで給飼パラメータなどを設定できる。

また、新規の酪農機器を本通信ネットワークに容易に接続できるように通信制御プログラムの一部を自動的に生成する開発環境を構築した。この開発環境は機器を制御するコマンドをテキストファイルに記述することにより通信制御プログラムの一部を自動生成する。これにより、コマンドの変更・追加がテキストファイルを編集することで済み、通信制御システムの開発を効率的に行うことができる。

1. はじめに

牛舎内には牛に自動的に飼料を与える自動給飼機、搾乳を行う搾乳機、搾乳量を計量する乳量計など省力化酪農機器がある。現在、それぞれは独立なシステムとして動作している。飼料の与え方で搾乳量に大きな違いが出ることから搾乳量のデータを収集し、きめ細かい給飼量の制御を行うことが求められている。

本通信制御システムは牛舎管理の省力化を図るため、独立に動作していた牛舎内の酪農機器を通信ネットワークでパソコンと接続し、遠隔制御、遠隔監視を可能にするものである。これにより、例えば、乳量計で計測した搾乳量をパソコンにデータとして蓄積したり、パソコン上で詳細に給飼パラメー

ータを設定し、自動給飼させることが可能になる。あわせて、新規の酪農機器の追加や既存の酪農機器における制御機能の追加、変更に関して容易に対応できる開発環境の構築を含んだ通信制御システムの開発を行ったので以下に報告する。

2. システムの構成

現在、自動給飼機、乳量計などの酪農機器は独立に動作している。自動給飼機は本体に付属する入力装置を用いて給飼パラメータを入力していた。この方式は対話型であるが数字ボタンと機能ボタンを組み合わせるため入力操作が複雑であった。また、乳量計は計測した乳量を手動で記録するものであり、集計などの管理に手間がかかっていた。

本通信システムはその構成を図1に示すように酪農機器とパソコンを通信ネットワークで接続するもので、パソコンからGUI (Graphical User Interface) を使用して自動給飼機

*北原電牧株式会社

の設定を行ったり、動作状態の監視を行うなど、使い勝手がよく、操作性が向上し、高度な制御を可能にする。また、監視カメラをパソコンと接続することにより、従来、巡回して行っていた牛の健康状態の監視を、母屋から遠隔で行ったり、温度のデータを収集することによって換気扇を制御することも可能となる。

本通信制御システムの特長を以下に示す。

(1) GUIによる操作性の向上

パソコンから集中管理するため、使いやすいユーザ・インターフェース (GUI) を用い、パラメータ入力や動作のモニタなどが可能である。

(2) 通信制御システムの高機能化

自動給飼機をはじめ各酪農機器はすべてパソコンから通信により制御し、各酪農機器間の通信は行わない。パソコンと複数の酪農機器との間の通信はRS-485を使用する。酪農機器の制御装置にはマルチタスク型のリアルタイムOSを搭載し、非同期の通信に対応できる。

(3) システムの拡張性

本通信制御システムにおいて、新たな酪農機器の追加や、既存の酪農機器の制御機能の変更・追加に関して、容易に対応できるような開発環境を有する。

本通信システムの制御対象となる自動給飼機を図2に示す。

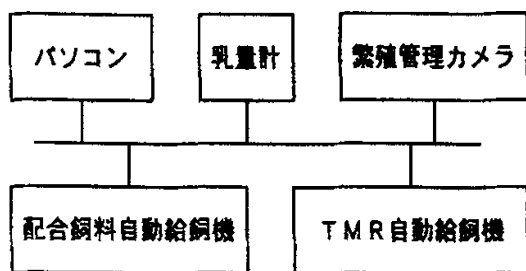


図1 牛舎内酪農機器通信制御システムの構成

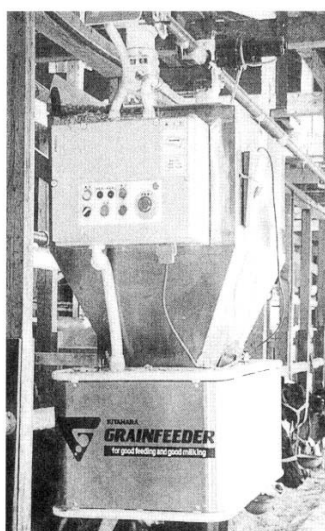


図2 自動給飼機

3. GUIによる操作性の向上

自動給飼機とパソコンとの接続により、従来キャラクタ・ベースで行っていた給飼パラメータの設定を図3に示すようにグラフィカルに設定できるようにした。

設定するパラメータとして、牛の頭数、搭載できる飼料の容量 (タンク容量)、給飼時刻、給飼する量に相当する給飼割合など給飼に関する基本的なデータを入力する。

次に、給飼する順序を指定する給飼パターンを設定を行う。設定はアイコンをマウスでドラッグして作成する (図4)。まずは、アイコンの並んだウィンドウから給飼時刻を設定するアイコンを選ぶ。12通り設定できる時刻から、これから設定する給飼パターンを適用する時刻を選ぶ。次に、給飼する牛を指定する給飼アイコンをドラッグして、給飼する牛の番号を指定する。必要ならば飼料を補給する補給アイコンをドラッグして、補給位置を指定するなどして給飼の方法を設定する。給飼アイコンと補給アイコンで給飼機の動作を指定する。このシステムでは天候などの諸条件に対応するため、給飼パターンを複数定義できる。

給飼パターンの設定後、自動給飼機にデータを転送する。その後、自動給飼機内のリアルタイムクロックにより、設定した時刻になると自動的に給飼を開始する。設定した各給飼

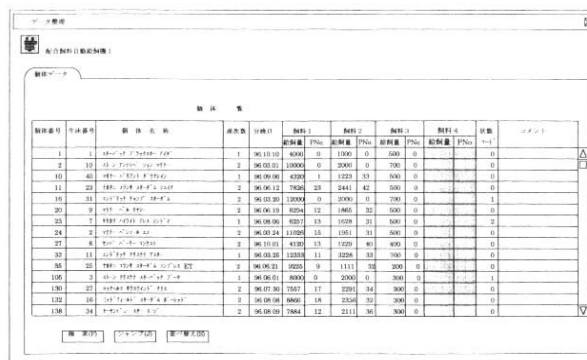


図3 給飼パラメータの設定

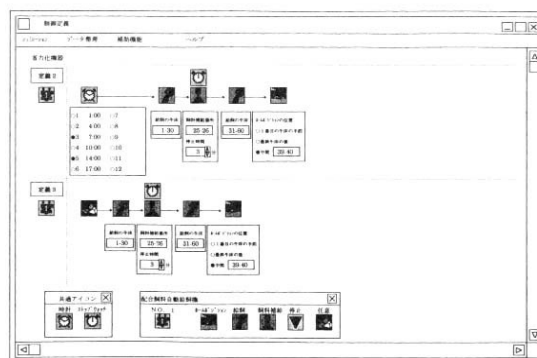


図4 給飼パターン設定

パラメータをメッセージとして送信するフローを図5に示す。

給飼機の動作モニタ画面では牛舎のレイアウトを示した図上に給飼機の位置や給飼状態（給飼中、待機中、移動中など）をリアルタイムに表示できるので遠隔監視が可能になる（図6）。

4. 通信制御システムの高機能化

4.1 事象駆動型リアルタイム OS

自動給飼機の制御装置には68000系のCPU(TMP68301)を使用し、シリアルポートの制御（送受信の制御）、I/Oポートの制御（給飼モータ、走行モータなど）を行う。CPUボードを図7に示す。

通信制御システムは基本ソフトとしてリアルタイム OS を別に開発した。このリアルタイム OS は割込などのイベントによって処理（タスク）の切替がおこる事象駆動型で、タスク切替時の優先度は固定である。この OS 上にタスクとして送信タスク、受信タスクの通信サービス、メッセージ解析タスク、給飼量算出タスク、給飼モータ制御タスクなどのアプリケーションを登録する。

リアルタイム OS のカーネルと各タスクの関係を図8に示す。各タスクはカーネル上で実行され、タスクの切替はカー

ネルがスケジューリングする。各タスクはカーネルに対するイベントの発行を意識してプログラミングするだけでよく、プログラムを機能的にモジュール化でき、それぞれ独立に開発できる。リアルタイム OS の仕様を表1に示す。

リアルタイム OS とメッセージ解析タスクなどのアプリケーションは最終的にROM化し、CPUボードに実装する。

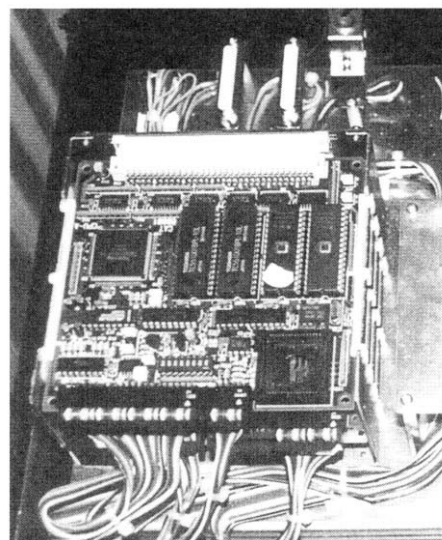


図7 CPUボード

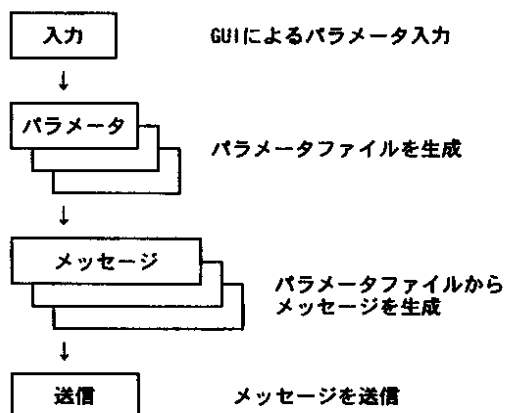
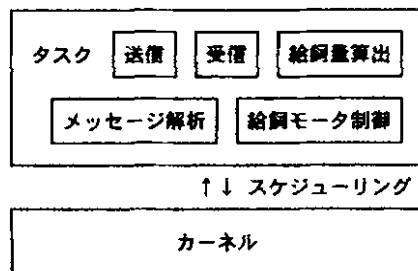


図5 設定したメッセージを送信するフロー



各タスクはカーネル上で実行され、カーネルによってスケジューリングされる。

図8 カーネルとタスク

表1 リアルタイムOSの仕様

タスク数	32 (最大)
登録イベント	104
タスクの切替	事象駆動、非プリエンティブ方式
タスクの優先順位	固定
カーネルサイズ	約4KB
主な機能	スケジューラ機能 タスク間同期通信機能 割込管理機能 事象発生管理機能 資源管理機能

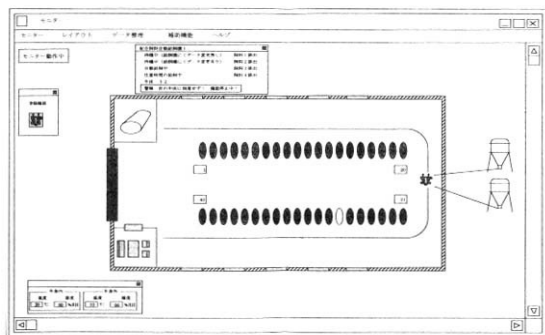


図6 動作モニタ

4.2 通信メッセージの仕様

パソコン上で作成された給飼パラメータや給飼パターンを自動給飼機に転送する際、機能ごとにメッセージに分割して送信する。例えば、牛の頭数、給飼時刻、タンク容量、給飼割合などそれぞれをメッセージとして自動給飼機に送信する。また、給飼機の動作状態をモニタする場合など、状態をモニタするメッセージを自動給飼機に送信し、自動給飼機から状態のデータを含んだメッセージを受信する。

メッセージはヘッダ部、コマンド部、データ部、フッタ部から構成される。ヘッダ部は送信先など通信に必要な情報を、コマンド部は、例えば、牛の頭数の設定などデータ部の種類を、データ部は牛の頭数など具体的な数値を、フッタ部はメッセージごとに付与される通信の情報を示す。メッセージの例を図9に示す。

ヘッダ	コマンド	データ	フッタ
010000C	CGWS-	H02	25

図9 メッセージの例

4.3 通信メッセージの流れ

酪農機器側において通信制御システムの基本的な処理の流れ以下に示す(図10)。

- (1)受信タスクがメッセージを受信バッファに格納し、受信終了イベントを発行する。
- (2)受信終了イベントにより、メッセージ解析タスクが受信バッファからメッセージ内のコマンドを読み込む。
- (3)メッセージ解析タスクはコマンドを解析し、給飼パラメータの書込、給飼モータの制御などを行う。また、動作状態などパソコンに送信する場合、メッセージを作成し、送信タスクに処理を依頼する。

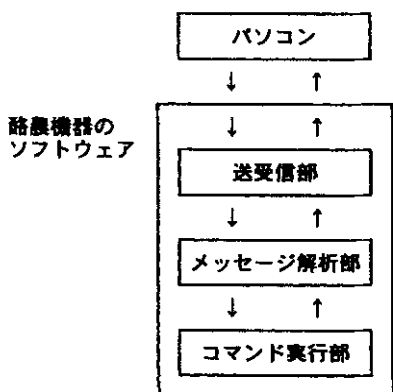


図10 通信制御システムの処理の流れ

5 システムの拡張性

本通信制御システムにおいて各酪農機器の制御機能の追

加、変更や、新規の酪農機器の追加の際に生じる通信メッセージの追加、変更を容易に行うため、通信制御システムのソフトウェアの一部を自動生成する開発環境を構築した。自動生成するため、各酪農機器に固有な部分と共通な部分を明確にした。メッセージ解析部のような解析手法は共通であるが、その内容が固有であり明確に区別できない部分は固有な部分を平易なテキスト形式で記述し、ソースファイルを自動的に生成する。

5.1 メッセージの解析

メッセージ解析は自動給飼機以外の酪農機器にも当てはまるものである。基本的に他の酪農機器の制御装置も自動給飼機と同じ68000系のCPUを使用するので基本ソフトウェアのリアルタイムOSには基本的に変更する必要はない。一方、メッセージ解析部については解析のアルゴリズムは同一でよいが、解析の結果、次に処理するタスクなどの関連づけが各酪農機器に固有のものとなる。

メッセージ解析部ではコマンド部分の解析結果を対応する関数やタスクに引数として渡す。メッセージ解析部はメッセージ解析部の初期化を行う初期化部、コマンドを解析し必要な処理を行う関数を呼ぶ解析部、最終的にコマンドで指定した処理を行う処理部から構成される(図11)。処理部はメモリのアクセスや他のタスクを起動するイベントを発行する。自動給飼機の場合、解析するコマンドの種類はおおよそ150である。

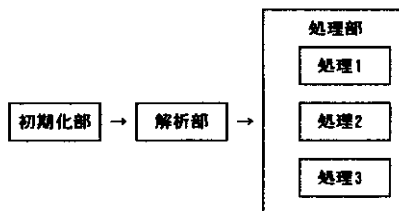


図11 メッセージ解析部の構成

メッセージ解析部は各酪農機器に対して、初期化部は共通の部分、処理部は固有の部分である。解析部はコマンドを解析して関数やタスクを呼ぶ仕組みは共通の部分であり、解析するコマンドが固有の部分であると考えられる。そのため、メッセージ解析部は固有の部分と共通の部分が混在していると言える。そこで、図12に示すような定義ファイルにコマンドと呼び出す関数の対応を記述し、その定義ファイルから解析部を自動生成する処理系を開発した。この処理系を用いると図13に示すように、酪農機器AとBの定義ファイルである定義A、定義Bを記述し、定義Aから酪農機器A用のメッセージ解析部を、定義Bから酪農機器B用のメッセージ解析部を自動生成する。

```
CGWT:rx_kyuji_time();/* 給飼時刻 */
CGWT1:rx_tank((int)e);/* 飼料1タンク容量 */
CGWT2:rx_tank((int)e);/* 飼料2タンク容量 */
CGJ--:status_monitor();/* 状態モニタ */
```

図12 メッセージ解析部の定義ファイルの例

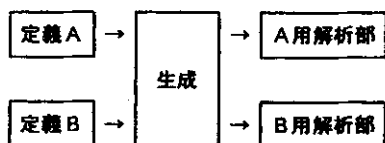


図13 メッセージ解析部の構成

5.2 メッセージ解析部の自動生成

メッセージ解析部を各酪農機器に固有の部分として考えると、新規の酪農機器用に新たにメッセージ解析部を作成しなければならない。また、すでに作成したメッセージ解析部に対して、コマンドの変更や追加に際し、該当部分を探して編集することは必ずしも効率的とは言えない。メッセージ解析部は解析したいコマンドとそれによって実行させたい関数を対応づけているだけなので、コマンドと関数の対応を平易に記述できればメッセージ解析部の作成を効率的に行える。例えば、試験用にコマンドを臨時に追加するような柔軟な対応が可能になり、システム開発の効率化が期待できる。

定義ファイルは図12に示すように、コマンド、関数、コメントの順に記述され、コロン(:)をフィールドセパレータとしている。この定義ファイルから自動生成部では受信したコマンドと定義ファイルに記述した文字列とを比較し、一致した場合、定義した関数を呼ぶ switch-case タイプのC言語のソース・ファイルを自動生成する。

メッセージ解析部を自動生成するためにコマンドの文字列をグループ化する。これを説明するために図14に示す定義ファイルの例を用いる。コマンドは5つの文字から構成される。最初の‘CG’は自動給飼機に対するコマンドを示しており、自動給飼機のメッセージ解析部にとって、最初の2文字は共通である。よって、残りの3文字をグループ化する。グループ化はまず、3文字目のグループに分ける。図14の例では3文字目が‘A’のグループに分けることになる。次に、3文字目の各グループごとに4文字目をグループに分ける。この場合、3文字目がAのグループで4文字目が‘D’、‘E’、‘F’の各グループに分けることになる。最後に5文字目で目的の関数に対応づける。グループ化したコマンドの様子を図15に示す。

グループ化した後、コマンドを順にソースファイルに置き換えていく。自動生成したソースファイルを図16示す(一部省略してある)。実質的には自動生成したソースファイルを参照する必要はないが、参照する場合、メッセージ解析部の内容がわかるように定義ファイルに記述したコメントを入れることができる。

これにより今後開発する他の酪農機器の通信制御部分に適用する際、コマンドと実行させたい関数を記述するだけで通信制御システムが生成できる。あとは酪農機器固有の動作部分の開発のみでよい。また、コマンドの変更・追加によるメッセージ解析部の変更・追加は定義ファイルの編集のみを行えばよく、開発の効率化を図ることができる。また、この開発環境は酪農機器側のみならずパソコン側のメッセージ解析部にも同様に適用できる。

```
CGADG:function_A();;
CGADH:function_B();;
CGADI:function_C();;
CGAEG:function_D();;
CGAEH:function_E();;
CGAEI:function_F();;
CGAFG:function_G();;
CGAFH:function_H();;
CGAFI:function_I();;
```

図14 メッセージ解析部の定義ファイルの例

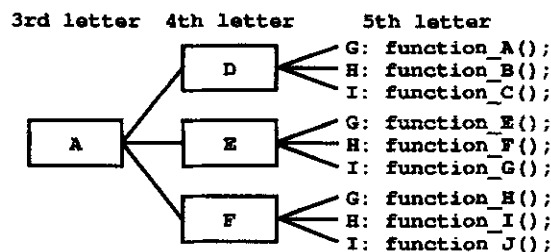


図15 コマンドのグループ化

```
switch(third=buf[3]) {
case 'W':
switch(fourth=buf[4]) {
case 'T':
switch(fifth=buf[5]) {
case '-': /* 給飼時刻 */
rx_kyuji_time();
}
}
}
}
```

図16 自動生成の結果

6. 通信制御システムの開発の結果

本通信制御システムの開発において、GUIの開発、GUIで設定された各パラメータをそれぞれメッセージとして通信するシステムの開発、酪農機器向けに通信のような非同期なイベントの発生にも対応できるようなリアルタイムOSの開発により、遠隔制御、遠隔監視が可能になり、酪農作業の省力化を図ることが可能になった。

本通信ネットワークへの新規酪農機器の追加や既存の酪農機器の制御機能の変更、追加によるメッセージの変更、追加を容易に行うため、メッセージ解析部を自動生成する開発環境を構築した。

GUIで設定した給飼パラメータをメッセージに分割して送信し、給飼機でそのメッセージを解析し給飼パラメータの設

定や給飼機の動作状態のパソコンへの送信などを行う機能が正常に動作したことを確認した。図 17 に実際の通信した送受信のメッセージを示す。これらのメッセージは給飼機の状態を問い合わせる送信メッセージ（1，3 行目 >）に対する給飼機からの受信メッセージ（2，4 行目 <）を示す。

```
> 0100009CGR5-10
< 000100CCGAS-H0233
> 0100009CGRT208
< 000100DCGAT2H1551F
```

図17 通信試験の結果（一部）

7. まとめ

複数の酪農機器をパソコンで集中管理できる通信制御システムを開発した。本通信制御システムにより母屋から遠隔制御、遠隔監視が可能になった。GUIの開発により、パラメータの設定などをパソコンの画面上でビジュアルに表示することで使いやすく操作性も向上した。また、制御機能の追加や変更、新規の酪農機器にも通信制御システムを容易に組み込める開発環境を構築した。これによって、本通信制御システムの開発の効率化が期待できる。

今後、通信エラーなどが生じた場合の対応部分の実装を行うとともに、本通信制御システムを他の酪農機器にも適用し、牛舎管理のシステムとして実証試験を行う予定である。

なお、本通信制御システムの開発は生物系特定産業技術研究推進機構（生研機構）の事業である、生研機構民間委託研究の大課題「牛の省力的安全放牧管理に関する研究開発」における、中課題「係留式牛舎における超省力飼養管理システムの研究開発」の一環として北原電牧株式会社と共同で行ったものである。

参考文献

- 1) sed & awk, Dale Dougherty, O' Reilly & Associates, Inc. (1991)
- 2) データ通信プロトコル入門, 永井正武, CQ 出版株式会社 (1991)