

琉球大学学術リポジトリ

自律型バッテリーレス太陽光発電システムの研究開発と農業用灌漑系への適用計画： インターネット遠隔計測・制御技術を用いた自然エネルギーシステムの構築

メタデータ	<p>言語: Japanese</p> <p>出版者: 玉城史朗</p> <p>公開日: 2009-07-28</p> <p>キーワード (Ja): 無線LAN, バッテリーレス太陽光発電システム, 揚水システム, 簡易型インターネットシステム, インターネット遠隔制御, 最適灌漑システム, インバータ周波数制御, 最適システム設計</p> <p>キーワード (En): provisional Internet System, Pomp up of Irrigation, Optimal System Design, Inverter Frequency Control, Internet Supervisory Control, Wireless LAN, Battery-less Solar-Power Generator, Optimal Irrigation System</p> <p>作成者: 玉城, 史朗, 吉永, 安俊, 名嘉村, 盛和, 屋我, 実, 瀬名波, 出, Tamaki, Shiro, Yoshinaga, anshun, Nakamura, Morikazu, Yaga, Minoru, Senaha, Izuru</p> <p>メールアドレス:</p> <p>所属:</p>
URL	http://hdl.handle.net/20.500.12000/11507

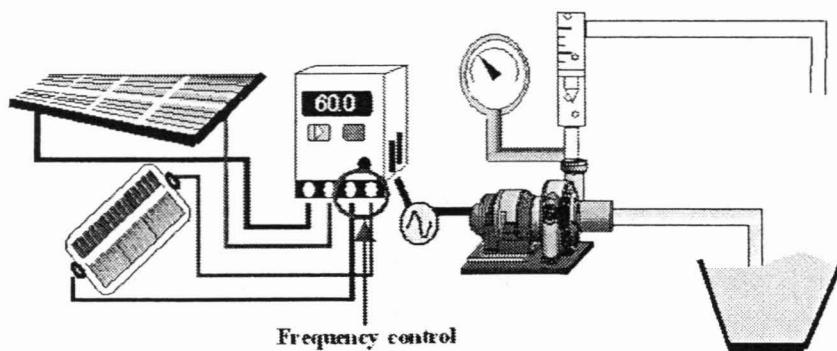


図 2.19: 太陽光発電揚水システムの概要図

システム駆動用太陽電池	129W, シリコン単結晶
周波数制御用太陽電池	350mA, 1.4V
インバータ	400W
揚水ポンプ	200W, 誘導電動機, 回転羽根型

表 2.3: システム構成要素の仕様

2.2 太陽光発電揚水システム

本研究では自然エネルギーの中でも盛んに研究・開発され、最も実用段階に到達している太陽電池を利用する。太陽光エネルギーは、太陽電池により直接電気エネルギーに変換することができるクリーンエネルギーであり、太陽がある限り電力を生産し続けることができるという点では半永久的に利用できる。太陽電池と共に研究が進んでいるものに風力発電機もあるが、太陽電池は風車のように動的部分がないので保守が容易であり、また、騒音問題を考慮する必要もない。通常、太陽電池から発電される電力はバッテリーに蓄えられる。しかし、バッテリーはコストが高く、過充電、過放電防止などのメンテナンスにより発生する労力も大きい。そこで、本研究では揚水システムにおけるメンテナンスをできるだけ少なくするため、バッテリーを使用せずに、太陽電池から出力される直流電力をインバータを介して交流電力に変換し、揚水ポンプを駆動させるシステムの構築を行う。

2.2.1 システムの概要

太陽光発電揚水システムの概要図を図 2.19、構成する機器の仕様を表 2.3 に示す。本システムは、太陽電池により発電した直流電力を一般的な自然エネルギー発電施設で使用しているようなバッテリーに蓄電せずに、インバータを介して直交流変換し三相誘導電動機を駆動させ、揚水ポンプにより水を汲み上げる仕組みである。誘導電動機の回転速度制御を周波数可変で行うために小型太陽電池をインバータに接続する。周波数は小型太陽電池

の出力に依存するため、インバータは日射量の変動に併せた周波数制御を行う。本システムでは、揚水ポンプの吐出口に手動バルブと圧力計を設置してある。バルブで流量を調整し管内の水圧を変えることで、水頭を変化させることができる。

太陽光発電揚水システムは、単に水を汲み上げ、タンクに貯蔵するのを目的としているため、一般の商用電源のような安定した電力供給を過度に重要視する必要はない。システムは、日射量により変動する太陽電池発電電力を直接揚水ポンプの電源としているため、日射量に応じた水の汲み上げ量を期待することができる。

揚水ポンプを駆動するには太陽電池からの直流電力をインバータにより直交流変換を行う。また、インバータの設定する周波数により揚水ポンプの回転数を決定する。周波数は通常、利用者が必要に応じて調整するが、本システムでは太陽電池出力特性に合わせて周波数を自動調整する。

例えば、日射量が少なく、太陽電池からの出力電圧が不十分な状態において、定格周波数 60Hz で電動機を駆動しようとすると、太陽電池の出力特性により太陽電池の電圧は低下し、電流は上昇する。しかし、電力をさらに必要とした場合、太陽電池は短絡状態になり、揚水システムへの電源供給も停止してしまう。

そこで、太陽電池の出力変動と同期して周波数も変動させることにより、この問題を解決する。ここで、変動する電源を用いるために、本システムにおいて考慮すべき点を以下に示す。

通常、モータを駆動するには入力電圧 V と入力周波数 f の比を一定にする必要がある。

$$\frac{V}{f} = \text{一定} \quad (2.1)$$

これは、モータ内部鉄心の時速密度 B を一定に保つためである。磁束密度 B は次式で表される。

$$B = \frac{V}{4kNfA} \quad (2.2)$$

それぞれの変数については以下のとおりである。

- B : 磁束密度 [T]
- V : モータ入力電圧 [V]
- k : 波形率
- N : コイルの巻数
- f : 周波数 [Hz]
- A : 鉄心断面積

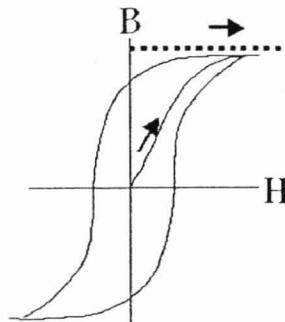


図 2.20: ヒステリシス曲線

上記2つの式から、磁束密度 B は V/f に比例することが分かる。また、磁性体の磁化のようすについて、磁性体に外部から加えた磁界 H に対して、磁束密度 B の関係を測定すると図 2.20 のヒステリシス曲線が得られる。

揚水システムで用いている誘導電動機の内部鉄心（磁性体）にはコイルが巻かれている。コイルに流れる電流が増加すると磁界 H も増加する。それに伴い磁束密度 B も増加するが、あるところまで磁束密度 B が増加すると磁気飽和現象となる。式(2.1)より、仮に周波数 f が一定でモータ入力電圧 V が増加する場合、磁束密度 B が増加することによって磁気飽和現象を引き起こし、結果的に誘導電動機に大電流を流してしまう。このような大電流はジュール熱を発生させ、コイル巻線を焦がす恐れがある。通常のインバータは入力電力が一定であるので、その出力周波数を設定して入力電圧 V と入力周波数 f の比を一定に保つことができる。しかし、本システムでは、インバータへの入力電圧は太陽光発電によるものであり、日射量により変動する。インバータはその変動に併せて出力周波数をその都度変化させ、モータへの入力電圧 V と入力周波数 f の比を一定に保つ必要がある。つまり、空が曇り太陽電池へのエネルギーが減少し、太陽電池の出力電力が低下したらインバータの出力周波数を減らし、ポンプの回転数を減少させる。逆に、空が晴れて出力電圧が上昇したら、インバータの出力周波数を増やし、ポンプの回転数を上昇させる制御が必要となる。そこで、制御用センサとして小型の太陽電池を用意し、インバータの出力周波数制御を行う。小型太陽電池をセンサとして用いることにより、周波数も太陽電池発電電圧と同等の挙動を示す。ここで、前提となるのは、センサに利用する小型太陽電池とシステム電力供給源の太陽電池の日射に対する特性がほぼ相似であるということである。

2.2.2 インバータ制御回路

我々はバッテリーを使用せずに太陽電池から供給される直流電力を直接インバータを介して交流に変換し、誘導電動機を駆動電源とする揚水ポンプを駆動し水を汲み上げる灌漑システムの構築を行った。本システムではシステム駆動用電源に太陽電池を使用しているので、商用電源を利用する場合では考慮の必要がないような特殊な事例が生じる。

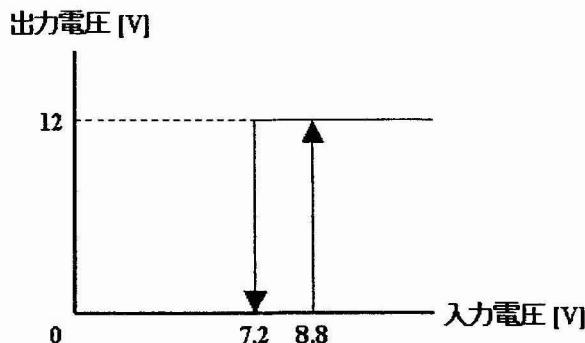


図 2.21: ヒステリシスによる出力特性

汎用インバータを商用電源で運転すると、負荷が大きく変化することがあっても電圧の変動が無いために安定した動作をする。本システムではインバータへの供給電圧は、太陽電池を用いているために日射量に依存する。すなわち、日射量が少ない条件下において消費電力が太陽電池の供給可能電力を上回ると、太陽電池の出力特性により太陽電池からの供給電圧は急激に減少し短絡状態に陥る。その結果、太陽電池からインバータへの供給電圧が不十分となり出力を停止する。

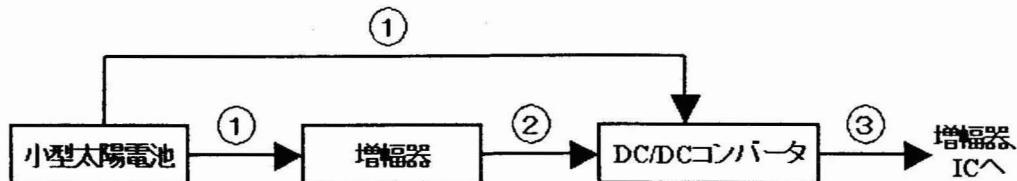
以上のような太陽電池、汎用インバータの諸特性による問題点を以下に示す。

- 市販の汎用インバータは太陽電池から充分な電力を供給されてもシステムの無人始動が不可能である。
- 曇りや誤作動などの影響で太陽電池電源からの電力の供給が一時的に停止した後、再度電力供給が行われた場合、システム再始動に必要なりセット（異常解除）、運転開始の動作を手動によって行わねばならない。

この2つの問題点は、太陽光発電揚水システムを設置するために解決すべき最重要課題である。これらの問題点を解決するために、インバータの異常状態を解除するためのリセット機能、自動運転開始機能および周波数制御機能を装備したインバータ制御回路の設計を行った。設計した回路はその機能により定電圧回路、周波数制御回路、異常検出回路、運転指令回路の4つの部分に分類することができる。以下にその各機能の動作原理を示す。

定電圧回路

回路で用いる増幅器の電源としては定格12V必要であり、その12Vを得るためにDC/DCコンバータ（7V～12V入力で定格電圧12Vを出力、以下コンバータ）を利用している。そのコンバータの入力は小型太陽電池から供給されている。日の出とともに小型太陽電池から出力電圧が非常にゆるやかに上昇していくと、コンバータは正常に動作しない。そこで電圧が低い間はコンバータに入力電圧を供給せずに、入力電圧が8.8Vに達した時に瞬時



1. 日射量に応じた電圧 (0~10V) を小型太陽電池から出力する。
2. 増幅器に入力された電圧が 8.8V を超えるとコンバータを動作させるための信号を発信する。
3. 増幅器からの信号を受けてコンバータから定電圧 12V を出力する。

図 2.22: 定電圧回路の動作原理

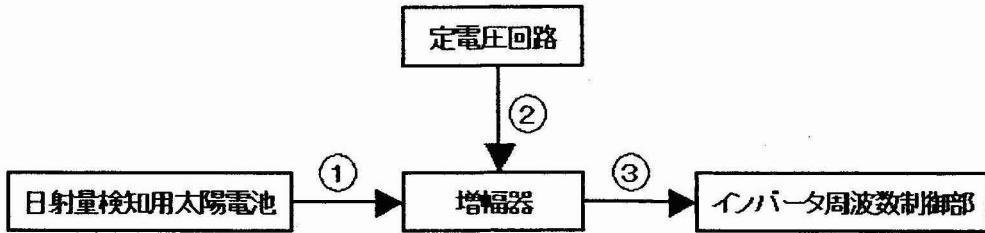
に電圧がコンバータに入力されるように回路を作製した。これにより、コンバータから正常に定電圧を取り出すことができる。ここで回路の ON/OFF が過敏にならないようにヒステリシスを持たせてある。そのため一旦コンバータから定電圧が出力されるとコンバータへの入力電圧が 7.2V 以下になるまで定電力を出力する。図 2.21 にヒステリシスによる出力特性、図 2.22 に動作原理を示す。

周波数制御回路

誘導電動機の回転数を可変させるために小型太陽電池からの出力を増幅器を通して増幅し、周波数制御を行う。増幅器入力部の手前に可変抵抗を直列に接続し、可変抵抗の抵抗値を変えることにより、どの程度の日射量でインバータの周波数制御部に信号を送るかを調節することができる。この可変抵抗は下記の運転指令回路にも同様に影響する。この周波数制御回路により太陽電池の出力、すなわち日射量の変動に応じて周波数を可変することができ、揚水ポンプの回転数を調整できる。また、日射量検出用太陽電池に適当な並列抵抗をつなぐことにより太陽電池の出力電圧を線形化することが可能となるため、周波数と太陽電池発電電圧、周波数と日射量に関し比例の関係となる。図 2.23 に動作原理を示す。

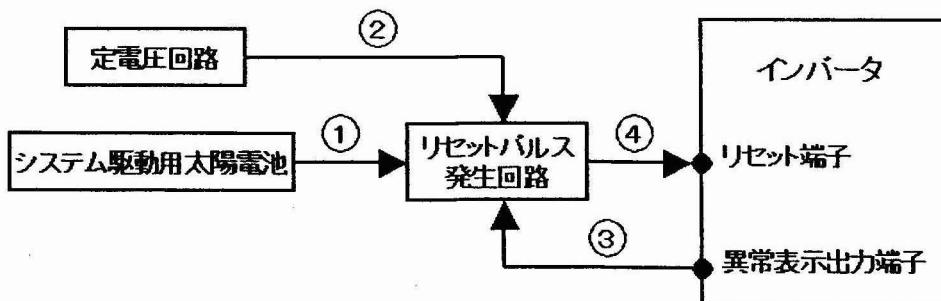
異常解除回路

汎用インバータは、日射量が低下し、入力電圧が規定値より下がると異常状態を画面表示し、接続している負荷への電力供給を停止する。インバータが一旦異常状態になるとリセット信号をインバータ側に送信しない限り、例えインバータへ運動信号が入力されても誘導電動機の運転は開始されない。そこで、インバータから異常信号が発信されるとリセットパルス発生回路を動作させ、リセット信号をインバータに送信するように設計を



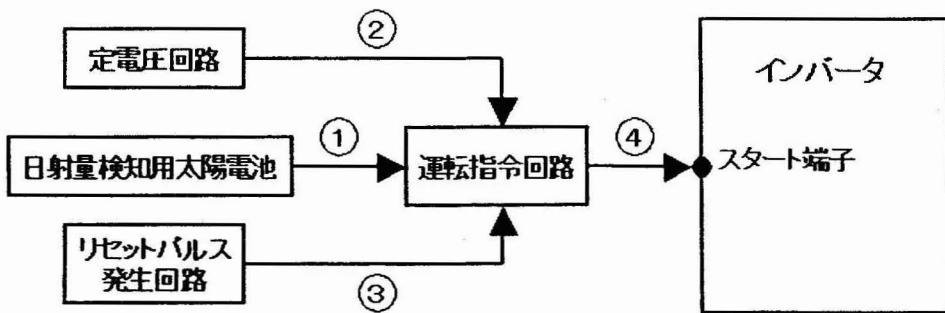
1. 日射量に応じた電圧 (0~1.4V) を日射量検知用太陽電池から出力する.
2. 増幅器を動作させるのに必要な定電圧 12V を出力する.
3. 日射量検知用太陽電池の出力を増幅器により 0~5V に増幅した電圧を出力する. インバータ側の周波数制御部に 0~5V を入力することにより、インバータの周波数制御部は 0~70Hz の周波数を出力する.

図 2.23: 周波数制御回路の動作原理



1. 日射量に応じた電圧 (0~300V) をシステム駆動用太陽電池から出力する.
2. リセットパルス発生回路の増幅器が動作するのに必要な 12V を出力する.
3. インバータへの入力電圧が規定値以下になると異常信号を発信する.
4. 異常信号入力後、充分な日射量により定電圧回路から 12V、太陽電池からの出力 280V 以上という条件を満たした時、1 パルスの信号が出力される。出力されたパルスはインバータのリセット端子へ入力され、インバータの異常解除が行われる。

図 2.24: 異常解除回路の動作原理



1. 日射量に応じた電圧 (0~1.4V) を日射量検知用太陽電池から出力する.
2. 運転指令回路の増幅器を動作させるに必要な 12V を出力する.
3. リセットパルス発生回路からリセット信号が出力される.
4. リセット信号がインバータに送られた後にスタート信号を出力する. リセットパルス発生回路は入力信号 Low レベルから瞬時に High レベルに立ち上がった瞬間に 1 パルスのリセット信号を出力する. 逆に運転指令回路は最初 High レベルの出力だが, 入力信号が入った瞬間 Low レベルになり, 1 パルス分の時間経過後, 再び High レベルに戻る.

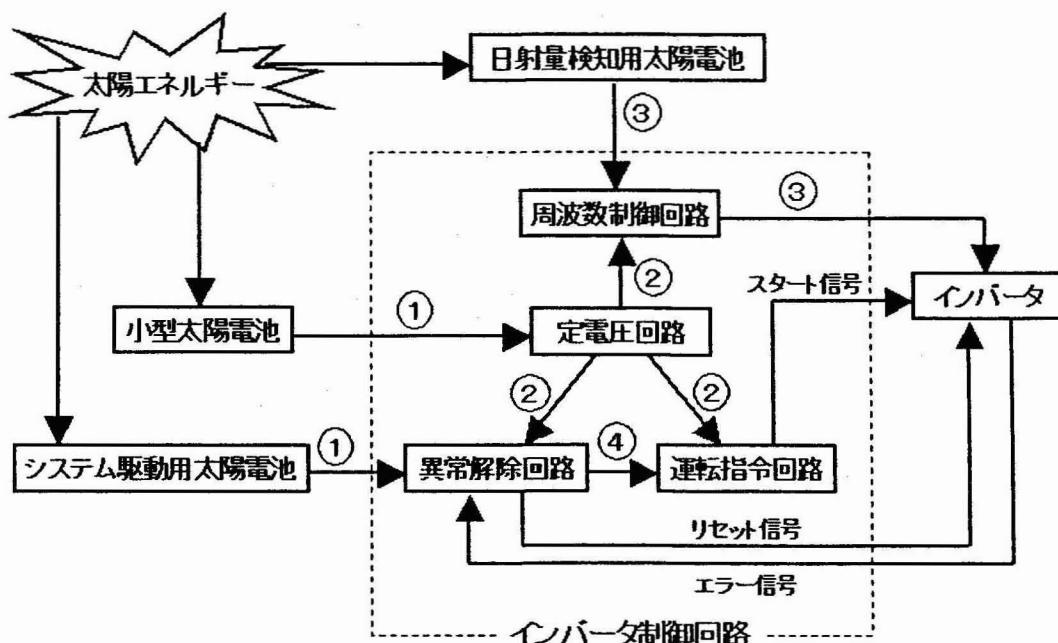
図 2.25: 運転指令回路の動作原理

行う. 定電圧回路から 12V, コンバータの出力電圧として用いているシステム駆動用太陽電池からの出力が 280V 以上であるという 2 つの条件を満たした時, インバータの異常状態を解除するためのリセット信号を発信する. このリセットパルス発生回路からリセット信号がインバータに送信されると, インバータは運転開始待ち状態になる. 図 2.24 に動作原理を示す.

運転指令回路

誘導電動機を運転するにはインバータ側からの交流電力の供給が必要である. 供給を開始するにはインバータの運転指令部に信号を送らなければならない. また, 何らかの原因でインバータが異常状態になり誘導電動機の運転が停止した場合, 異常状態を解除するためのリセット信号をインバータが受信し運転開始可能状態になる. しかし, 運転を再開するためには運転指令部に信号を送信しなければならない. そこでコンバータが正常な電圧を出力し, 異常解除回路がリセット信号を送信した後に, 運転指令信号がインバータ側に送信され揚水ポンプの運転が開始される. 図 2.25 に動作原理を示す.

これら 4 種類の機能を持つインバータ制御回路を開発することにより, インバータが異常状態に陥った場合でも, リセット信号をインバータに送信しエラー解除を自動的に行



1. 日射量に対し非線形な電圧を発電する。
2. 増幅器、ICに必要な定電圧を供給する。
3. 日射量に対し線形な電圧を供給し、周波数を制御する。
4. リセット信号をインバータに送信し異常状態を解除したことを伝えた後、運転指令回路からスタート信号を発信することでインバータを自動起動する。

図 2.26: インバータ制御回路概略図

う。そして、再始動するためのスタート信号を送信し、自動的に運転を開始する。これにより、無人状態における太陽光発電揚水システムの完全自動運転化を実現する。インバータ制御回路の各機能が連携した内部動作フローチャートを図 2.26 に示す。

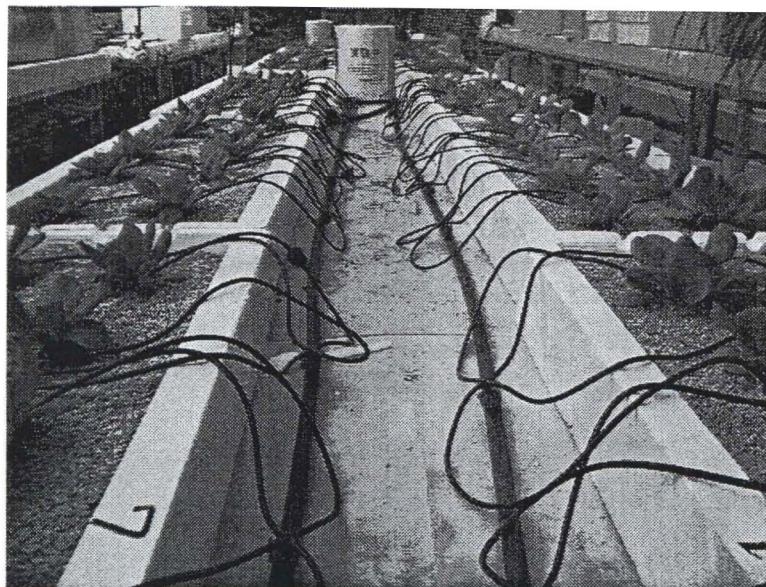


図 2.27: 灌水システムによる作物への灌水

2.3 灌溉システム

灌溉システムはポンプ、電磁弁、環境データ取得や電磁弁制御を行うコントロールボックスにより構成されている。灌水システムは、栽培環境条件に応じて変化する作物の必要水分量を求めるために必要な温室内の気温、湿度、日射量や土壤水分量などの環境データをセンサ[13]より取得し、エージェントへと渡す[14]。また、エージェントからの灌水命令を基に散水弁を制御し、作物が生育する上での最適水量を給水する。灌水システムとエージェントシステム間のデータの送受信にはソケット通信を利用する。

2.3.1 灌水方法

灌水システムによる作物への灌水のようすを図 2.27 に示す。灌水は流量制御ペグを作物の根元付近の土中に挿し、流量制御ペグの穴から水を給水することで行う。流量制御ペグはチューブ内に一定の水圧をかけることにより、単位時間に一定量給水する。作物に任意の水量を給水したい場合、単位時間あたりの給水量から給水時間を求め、灌水すればよい。

一定の水圧をかけるためのポンプを図 2.28 に示す。ポンプは、給水が始まるとアキュムレータ内に蓄圧された圧力が低下すると、圧力センサが動作しポンプが自動的に始動する。また、給水量が減少すると流量スイッチが動作し、ポンプが自動的に停止する。ポンプはこの一連の動作を繰り返す。

給水制御には図 2.29 の電磁弁を用いる。電磁弁は流量制御ペグとポンプとの間に設置されている。電磁弁に電圧をかけている間は給水し、電圧をかけない間は給水を停止する。電圧の ON/OFF はコントロールボックスで制御する。



図 2.28: ポンプ

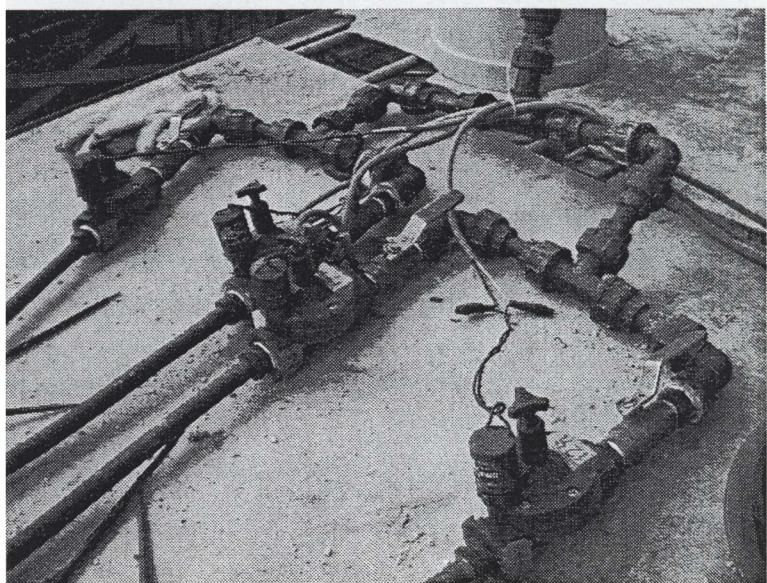


図 2.29: 電磁弁

2.3.2 コントロールボックス

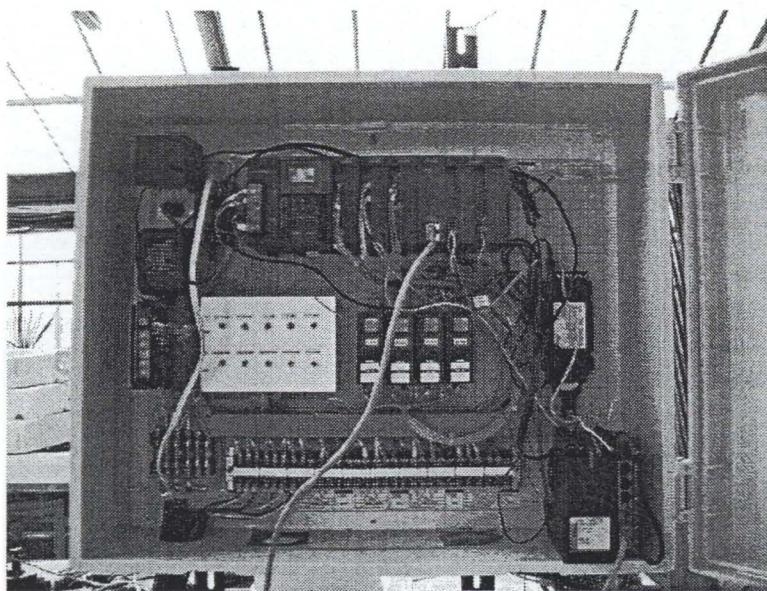


図 2.30: コントロールボックス

コントロールボックスとは、農場に設置する精密機器を1つのボックスに収めたものを指す。コントロールボックスを図2.30に示す。コントロールボックスはシーケンサ (KV-700)を中心とした構成となっており、その他にも電源、演算器、HUB やリレーなどが収められている。

シーケンサ (KV-700) とラダープログラムについて

シーケンサとは、CPUとメモリを内蔵したFA (Factory Automation) 用の制御装置である[13]。シーケンサを用いることで、装置や操作盤に設置した様々な機器 (センサやスイッチ) からの信号を入力回路で取り込み、あらかじめプログラムされた条件で出力回路をON, OFFするなどのさまざまな制御が可能である。シーケンサにはKEYENCEのKV-700を用いている。

KV-700はKV-700シリーズの中の基本ユニットであり、他のKV-700シリーズのユニットを増設することで機能を拡張することができる。例えば、電力の状態を監視・制御するAC電源ユニット、センサを接続してデータ（入力電圧）を取得する拡張入力ユニット、制御する機器を接続する拡張出力ユニット、その他の機能を有する拡張特殊ユニットがある。本研究では拡張特殊ユニットであるイーサネットユニットを組み込んであるので、KV-700とインターネットを介した通信が可能となっている。

プログラムはWindows用の専用アプリケーションソフトKV BUILDERを使用して作成し、KV-700に転送・コンパイルすることで動作する。プログラム開発言語は一般的な

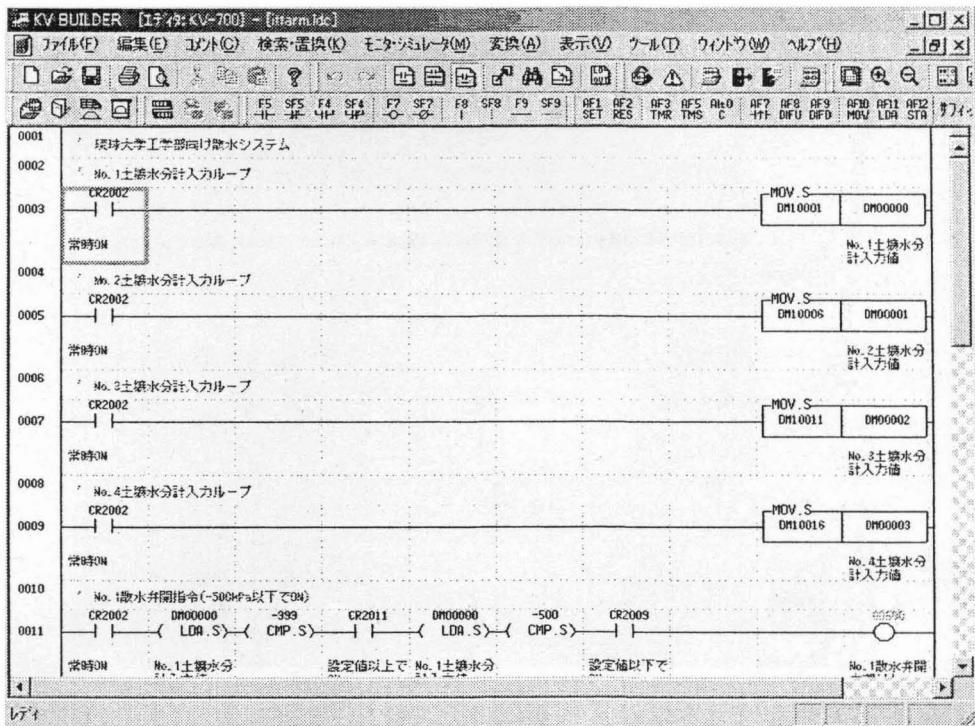


図 2.31: ラダープログラム開発のようす

C 言語や Java を用いることができず、ラダープログラムという専用の言語を使用する。ラダープログラム開発のようすを図 2.31 に示す。

ラダープログラム開発環境ではエージェントプログラムのような複雑なプログラミングが困難であるため、KV-700 が他のシステムで動作する C 言語や Java で開発されたエージェントプログラムと協調をはかることは難しい。そこで、ラダープログラムに用意されている KV ソケット通信の機能を用いて、エージェントシステムとのデータ通信を実現する。KV ソケット通信とは、イーサネット上の機器と TCP/IP または UDP/IP プロトコルにより任意のデータ送受信を行う機能である。パソコンやワークステーションだけでなく、イーサネットに対応しているさまざまな機器との通信が可能である。もちろん、ソケット通信を実現するためには、エージェントシステムに KV-700 と通信するためのソケットプログラムを組み込む必要がある。

2.3.3 ソケット通信

ソケットの由来

ソケットとは、TCP/IP を利用するアプリケーションのための API (Application Program Interface) を指し、UNIX に TCP/IP が実装された際のアプリケーション間通信の概念として考えられたことが始まりである [7]。1983 年に ARPA (米国国防総省の高等研究計画局: Advanced Research Projects Agency) の支援を受けて作られた 4.2BSD(Berkeley Standard

ソケットのタイプ	プロトコル名	特徴
ストリーム	TCP	信頼性が高く、送受信のデータ順序が保証されている。単一の接続で双方向通信が可能。
データグラム	UDP	信頼性は低いが、即時性が高い。送受信のデータ順序は保証されない。

表 2.4: ソケットのタイプ

Distrbution) に TCP/IP が実装されたことにより、プログラムから TCP/IP を使う API として用意された。ソケットとは OS およびそれに付随するソフトウェアが提供する API のことで、ソケットインターフェースという呼び方もある。

TCP と UDP

ソケットのタイプを表 2.4 に示す。

ソケットを説明するうえで必要となる IP 上で使用するプロトコルは ISO(International Standards Organization) の 4 層にあたる。インターネットプロトコル群では TCP と UDP が多く使用され、ネットワークのプログラムを作成する場合において使い分ける [8]。TCP はコネクション型のプログラム、UDP はコネクションレス型のプログラムを作成する場合に多く使用される。

コネクション型プログラムとはデータを送受信するに当たり、まず 2 つのプログラム間で接続を確立する。それぞれのプログラムの識別情報はこの接続確立時に交換され、以降はこの 2 つのプログラム間だけで通信が行われる。

コネクションレス型プログラムとはデータ送受信するたびにプログラムの識別情報が付加されており、データの送受信の回数が非常に少ないプログラムを作成する場合に使用される。

また、TCP は必ず特定の相手とのコネクションというものを意識するのに対し、UDP は特定の相手を持つ必要がないので、マルチキャスト通信をすることも可能である。しかし、TCP はコネクションを意識しているため、プロトコルでパケット消失などのエラー訂正を行うのに対し、UDP はコネクションレスなのでデータが本当に相手に届いたかを保証しない。そのため UDP の場合、データの信頼性は上位層で確保しなければならない。したがって、作ろうとしているプログラムの性格にもよるが、実装の容易さを優先する場合は UDP を使い、ネットワークの基幹に位置するような通信プログラムは信頼性のある TCP を使うことが多い。

クライアント/サーバ

ソケットの概念を使って通信を行うプログラムを作成する場合、2 つのプログラムを作成する。同じプログラムを 2 カ所で動かすのではなく、サーバとなりうるサーバプログラム

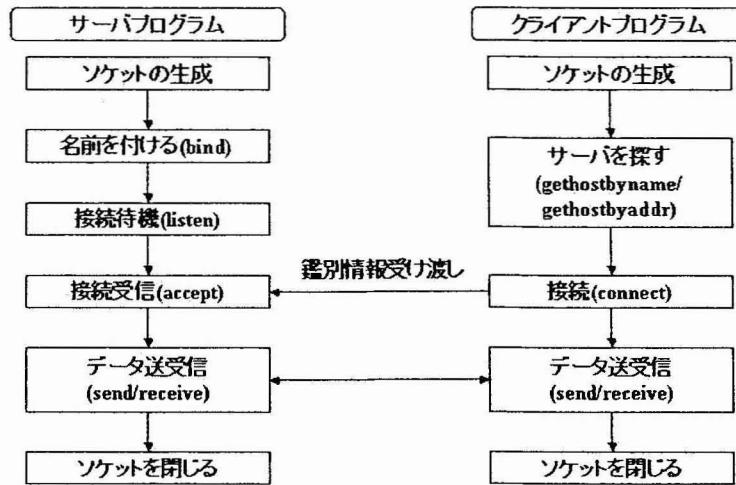


図 2.32: コネクション型プログラムの流れ

ムとクライアントになりうるクライアントプログラムの別々のプログラムを作成する。

サーバプログラムはソケットを用意して待ち、もう一方のクライアントプログラムが必要に応じてサーバのサービスを使用するためにソケットを用いて接続する。すなわち、サーバプログラムは、いつでも通信を受け付ける準備で待機する。クライアントプログラムがサーバプログラムに接続要求を出した段階でネットワーク通信がはじまる。

コネクション型の通信の場合は、サーバプログラムがクライアントプログラムからの接続要求を受け入れることで、仮想回線が確立される。その後はプログラムの流れによって決められたプロトコルに従って双方からのデータ送受信を行うことになる。

このプロトコルは「OSI参照モデルの第5層（セッション層）」にあたり、よく知られているインターネットプロトコルではFTP, Telnet, HTTP, NNTP, SMTPなどがある。TCP/IPでは両端の接続が保証されているので、そのプログラムの目的に沿った扱いやすい独自のプロトコルを作成してもよい。

5層以上のプロトコルを作成することにより、TCP/IPが実装されているプラットフォームであればソケットの概念は同じなので、メインフレームからモバイルPCまで、どんなコンピュータ同士でも通信が行えるプログラムの作成が可能である。

コネクションレス型の場合、接続の確立はしないので、最初にデータの送信を開始する側がクライアントになる。その後は上位のプロトコルにしたがって送受信が行なわれる。UDP上で動作するプロトコルとしては、DNS, BootP, TFTP, SNMPなどがある。

コネクション型プログラムの通信の流れ

コネクション型プログラムはデータを送受信する順番が決まっていないプログラムに多く用いられる。データを送信、受信するためにはまずサーバプログラムとクライアントプ

ログラムの接続を確立しなければならない。コネクション型プログラムの流れを図 2.32 に示す。

まず、サーバプログラムでソケットを作成し、それに名前を付ける (bind)。ソケットに名前を付けるときに、使用するポート番号とプロトコルを決定する。その後はクライアントから接続要求を待つ (listen)。クライアントから接続要求がきたら受け入れる。これで、サーバプログラムはいつでもクライアントプログラムからの接続要求を受け入れる (accept) ことが可能となる。

クライアントプログラムはソケットを作成し、サーバのホスト名またはIP アドレスを探す (gethostbyname/gethostbyaddr)。それからいよいよサーバへ接続要求をする (connect)。この時点で、サーバ側とクライアントの両側の識別情報が確認されたので、サーバとクライアントの間に一本の通信路が確立される。このことを仮想回線 (virtual circuit) の確立という。

このあとは両方のプログラムがソケットに対し、送受信する (send/receive) ことができる。送受信が終了したら両方のプログラムでソケットを閉じる。

なお、ソケットを閉じるタイミングは通常、クライアントプログラムからソケットを閉じる。サーバープログラムへは FIN パケットを送られ、サーバープログラムはクライアントプログラムがソケットを閉じたことをがわかる。