

修士論文

コンピュータによる作物の  
生育環境の計測・制御

岩手大学大学院教育学研究科  
修士課程教科教育専攻技術教育専修

32308027

菊池拓己

2010 年 3 月

## 目 次

第1章	序論	1
第1節	研究の背景	1
第2節	研究の目的	3
第2章	土壌の水分状態に応じた自動灌水制御	4
第1節	土壌中の水分状態	4
第2節	Arduino	9
第3節	制御方法	11
第3章	リモート灌水制御	17
第1節	システム概要	17
第2節	Arduino Ethernet Shield	18
第3節	Processing	19
第4節	制御方法	21
第5節	実行結果	24
第4章	ネットワーク計測	25
第1節	システム概要	25
第2節	計測方法	28
第3節	計測結果	29
第5章	画像計測による土壌酸度の測定	30
第1節	実験の目的	30
第2節	ImageJ	31
第3節	手順	32
第4節	結果	40
第5節	考察	42
第6章	結論	43
参考文献および web サイト		
謝辞		
付録資料		

# 第1章 序論

## 第1節 研究の背景

作物の生育にかかわる計測項目として、気温、地温、日射量、湿度、土壌水分などが挙げられる。従来は、これらの計測項目の観察および記録は主に手作業によって行われ、頻繁に行うことは困難であった。

しかしながら、フィールドサーバに代表される情報技術を活用した監視システムの導入によって作物の生育環境の計測や病害虫の発生などの圃場モニタリングが容易に行われるようになった。小型のデータロガーが安価で入手できるようになったことから気温や湿度等を一定時間ごとに計測・記録を行うことも多く行われるようになった。このように、作物の生育環境の計測は、様々な項目について日常的に行われ、栽培管理に利用されている。

一方で、コンピュータ制御を用いて作物の生育環境を調節すること、例えば日射量の調節や温度制御、灌水制御などは以前から一部で行われてきたが、システムが大掛かりになり費用もかかることから部分的な導入に留まっている。簡易な栽培管理を行うものに、家庭菜園の灌水制御があり、市販の自動灌水器を安価で購入できる。このタイプの灌水器は、灌水時間を設定するタイマー制御が用いられているが、タイマー制御の場合、水分過多が生じて作物の生育が阻害される恐れがある。水分センサが付属されて土壌の水分状態を計測できる機能を有する自動灌水器もあるが、センサの計測値と灌水時間や灌水量とが相互に関連して作物の生育に好適な土壌の水分状態を保持するようなものではない。これに関連して、愛知県農業総合試験場では、土壌の水分状態を測定し、生育ステージにあった灌水が行えるシステムを開発し、トマトの栽培を試みた結果、積算灌水量がタイマー制御を用いた灌水よりも少なく、それと同等の収量がえられたことを報告している<sup>1)</sup>。

教育現場における作物の生育管理は、未だ手作業による所が多く、コンピュータ技術を導入しているケースはほとんどみられないのが現状である。しかしながら、夏期休暇の栽培の水管理など、昨今の教員の職務の多忙化を鑑みて、コスト面も含めて簡易に導入できる計測・制御システムの構築が望まれる。従来の栽培学習へのコンピュータの利用は、ほとんどの場合、計測結果を表計算ソフトウェアでデータ処理したり、撮影した画像をレポート作成やweb ページ製作時に挿入することに留まっている。生物育成とコンピュータとの利用を図るうえで、アプリケーションソフトウェアの操作方法だけでなく、計測・制御への積極的な活用はほとんど行われていない。

さらに、平成20年版中学校学習指導要領技術・家庭編<sup>2)</sup>において、現行(平成10年版)の学習指導要領<sup>3)</sup>では「A 技術とものづくり」の中で選択して履修させる項目の1つとして位置づけられている「(5) 作物の栽培」が、「C 生物育成に関する技術」として必修化される。同様に、「B 情報とコンピュータ (6) プログラムと計測・制御」が、「D 情報に関する技術」の一部として必修化されるなどの内容構成の改善が図られた。「作物の栽培」が「生物育成」となったことにより、植物のみならず動物や水産生物も題材として取り扱うことができるようになったが、適切な育成環境の維持・管理技術を学習することにおいては共通しているといえる。そのための視点として、従来のコンピュータを利用したデータ整理やプレゼンテーションに加えて、例えば、リモート計測、リモート管理、画像による計測と診断、生物育成の時間スケールの変換など多面的で新規性に富んだコンピュータの使用を図ることが有効であると考えられる。

このような生物の育成環境の観察や管理を行うことによって出てきた新たな必要性や要求される計測項目そして問題点を、情報技術の助けを借りて解決する能力を育むためにベースとなるプログラミングの作成能力も必要である。幸いなことに、最近では高機能なフリーソフトを利用できるようになり、なかでも計測・制御関連ではAVR ワンチップマイコンである Arduino を用いた極めて優れた Processing

統合開発環境が利用できる<sup>4)</sup>。

本研究では、教育現場で容易に導入でき、作物の生育管理が簡単な Arduino を用いた Processing の環境を使用して、生物育成をターゲットとした情報技術の融合と展開を試みた。

## 第2節 研究の目的

教育現場における作物の生育へのコンピュータの利用を考察するにあたり、次の2点を柱に掲げた。それらの具体的な目的や方法および対象を図示したものが図1.1である。

- (1) 作物の生育環境の維持・管理の省力化

(2) 学習支援ツールとしてのコンピュータ活用

まず(1)に関して、かつて学校現場にコンピュータに導入されたことによって教職員の書類管理などの事務作業にかかる労力や時間が格段に効率化されたことが明らかなように、作物の生育の維持・管理にかかる人的負担をコンピュータの計測および制御技術を活用することで軽減することをねらいとし、前者は土壌の水分状態、気温、湿度などの生育環境を計測対象とし、後者は土壌の水分状態に応じて自動で灌水を行うシステムと遠隔操作による灌水システムの構築を目的とする。また、より実用的にするために、自動灌水システムを除き、すべてネットワークを介してリモートで操作できるよう設計を行った。

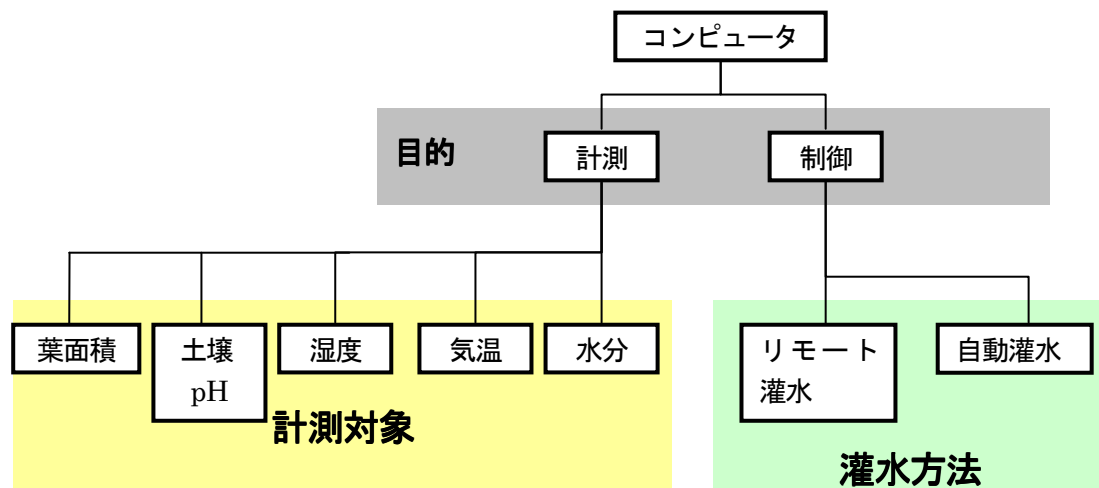


図 1.1 システムの目的および方法

次に、(2)に関しては、古くから CAI (Computer Aided Instruction) システムにみられるように学習者の能力や理解度に応じて、個別に学習を進められるものが知られている。CAI は、ドリル学習などの学習者の自習用プログラムとして用いられる。しかしながら、本研究で意図するコンピュータ活用は、対象となる生育環境の正確な把握および継続的な記録を可能にするための補助ツールとしての役割を指す。濃度などの人間の感覚量に依存する部分や気象条件などの一定間隔での観察・記録がその一例である。さらに、コンピュータを活用するメリットとして、計測データの蓄積や異なる計測対象の計測結果の比較などをもとにして生物の育成環境について考察することができる。そこで本研究では、図 1.1 に示す計測対象のうち、リモートでの土壌水分の計測および画像解析による土壌 pH の測定を行い、その解析結果および手法について評価することとした。

## 第2章 土壌の水分状態に応じた自動灌水制御

### 第1節 土壌中の水分状態

土壌中には水蒸気や水としての水分も存在するが、最も重要な水分は液体としての水である。土壌水には、土壌中の無機成分、有機成分、二酸化炭素などが存在し、実際には土壌溶液となっている。

土壌水には、以下のようなはたらきがある。

- 1) 植物に吸収利用されて、生育を促進させる
- 2) 土壌の成分を溶かし出すことで、植物に必要な養分を供給する
- 3) 土壌の微生物や動物の生活を支え、活性化する
- 4) 高温時には蒸発に伴って多量の熱を奪い、低温時には氷結して放熱することにより、地温の急激な変化を抑制する
- 5) 水は表面張力と凝集力が比較的大きいため、土壌孔隙中に保持されやすく、また土壌中を移動しやすいため、植物根に吸収されやすい。

これらのはたらきをもつ土壌水を量的に表す方法として、含水比（単位重量の土が保持する水分量）や水分率あるいは体積含水率（単位容積の土が保持する水分量）が一般的に用いられる。しかし、これらはいずれも土壌水を全体として捉えた量である。例えば、砂質土壌と粘質土壌と比較した場合、同じ水分量であっても、植物生育にとって前者は適量であり、後者では不足することがある。このような現象を説明するうえでは水分量以外に水の質的側面を示す物理的概念、すなわち土壌水分ポテンシャルが必要となる。土壌水分ポテンシャルは、次の4つの総和として表わされる。

#### 1) マトリックポテンシャル

土壌の表面近くでは、粒子の配列（マトリックス）が作った孔隙によって、毛管力・分子間力・クーロン力がはたらき、水分子と土壌粒子の間に相互作用が生じて水が保持される。

#### 2) 重力ポテンシャル

地表面近くの水は深い所にあるそれと比べると位置エネルギーが高い。このような高さの差から生じる水のポテンシャル。

#### 3) 浸透ポテンシャル

土壌水には様々な物質が溶けており、純粋との間に浸透圧が生じて水ポテンシャルは低下する。

#### 4) 粒度D

粒度を  $D_{cm}$  としたとき、 $pF = -\log D$  による capillary tension で扱っている。

このように、土壌の水分ポテンシャルには多くの要素が関係するが、通常の土壌状態には主としてマトリックポテンシャルが主成分となる<sup>1)</sup>。マトリックポテンシャルを  $cm$  で換算し、その絶対値を対数表示したものが  $pF$  ( $p$  は対数表示、 $F$  は自由水のエネルギーを表す) 値と呼ばれ、式 2.1 で与えられる。

$$pF = \log H \quad (\text{式 2.1})$$

$H$  : 水柱の高さ (cm)

また、単位をパスカルに換算すると

$$pF = \log(-10.2\phi) \quad (\text{式 2.2})$$

$\phi$  : 負の圧力 (kPa)

pF 値は、1935 年にスコフィールド (Schofield) によって提案され、簡単に土壤の水分状態を表現することができるようになったため、現在でも農業分野では広く使われている指標である。式 2.1、2.2 からわかるように、例えば pF=4 では  $H=10^4\text{cm}$  で  $-10\text{atm}$  に相当し、粒度は  $D=10^{-4}\text{cm}$ 、相対湿度は 99.5% である。pF 値が高いほど土壤が乾燥状態であることをあらわす。

pF 値の測定方法は、目的とする測定範囲に応じて表 2.1 に示すように幾つかに分けられる。本研究では、設置および測定の容易さ、測定範囲などを考慮し、テンシオメーター法の pF メータを使用した。

表 2.1 土壌水分ポテンシャル測定法<sup>2)</sup>

名 称	方 法	測定範囲(pF)
砂柱法	細砂上に試料を置き、水位を変えて水分を平衡させる方法。室内法、水位差は砂の径にもよるが 60cm までは可能。	
加圧法	耐圧容器内で加圧して試料水分を脱水平衡させる方法。特殊な素焼き板やセロファン膜が用いられる。	1.0～4.2
蒸気圧法	デシケーター内で既知蒸気圧の各種溶液と試料を平衡させる方法。	4.2～7
サイクロメーター法	乾湿球法または露天法に基づく微小温度降下を熱電対で測定し、平衡相対湿度から全ポテンシャルを求める方法。野外では低精度。	3.2 以上
テンシオメーター法	水を満たした素焼きカップを土壌中に埋設し、土壌に吸引される水負圧を測定する。野外測定可能。	2.8 付近まで
セラミック水分計法	一定孔隙径をもつセラミック片に電極を近づけて土壌に埋設し、孔隙径あたり pF 水分以下では電流 on、以上では off となることを測定原理。野外測定可能。	1.0～ 3.0 前後
ピエゾメーター法	たん水下あるいは地下水面下の飽和土層から、水の水圧をパイプで取り出し水柱高を測定する。	正圧域

使用した pF メータの基本的な構造を図 2.1 に示す。この測器は、セラミックでできた多孔質カップを真空ゲージに繋いだもので、内部はすべて水で満たされている。測定する土壌の中にこのカップを埋設すると、カップの中の水はセラミック壁の間隙を通して土壌水と接し、平衡に達する。一般に、土壌中に設置した当初は、テンシオメーター中の水は大気圧と等しい。一方、土壌水の圧力は、通常大気圧より低いので、カップ内の水に吸引圧が作用する。テンシオメーターは気密であることから、このとき、一定量の水が吸収され、テンシオメーター内の圧力は大気圧以下に下がる。この圧力低下分を真空ゲージで指示するようになっている。



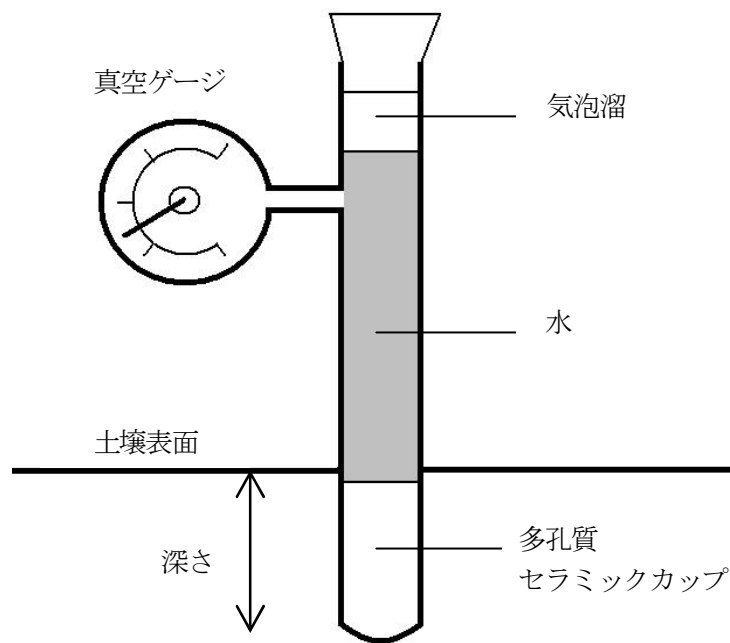


図 2.1 pF メータの基本的構成



図 2.2 使用した pF メータ（大紀理化工業株式会社製 DIK-8343）

表 2.2 仕様

受感部	素焼寸法	$\phi 8\text{mm} \times \text{L}50\text{mm}$
	受感部材質	素焼き（セラミック） アクリル樹脂
	埋設深さ	10cm
	全体寸法	$\phi 11\text{mm} \times \text{L}175\text{mm}$
	付属品	ウエスター栓 1 個
表示部	受感部接続方式	カプラー式
	指示方式	ゲージ圧
	測定範囲	pF0~2.7
	精度	$\pm 1.5\% \text{F.S. (at } 23 \pm 5^\circ\text{C)}$
	外径寸法	$\phi 67\text{mm} \times 32\text{Hmm}$ （突起部含まず）
	重量	約 105g

## 第2節 Arduino

本研究では、開発環境にイタリア製のデジタル I/O ボードである Arduino を使用した (図 2.3)。Arduino は、Atmel AVR マイコンというワンボードマイコンが組み込まれた入出力ボードで、14 チャンネルのデジタル入出力ピンと 6 チャンネルのアナログ入力ピンがある。アナログ入力の分解能は 10 ビットで、プログラムで設定することによってデジタル入出力ピンのうちの 6 チャンネルを PWM として使用することが可能である。プログラムは USB 経由で書き込みされるため、FTDI FT232RL チップが表面実装されており、USB 端子が付いている。電源は、通常 USB 接続によって 5V が供給されるほか、外部電源端子にバッテリーや AC アダプタを接続することも可能である。

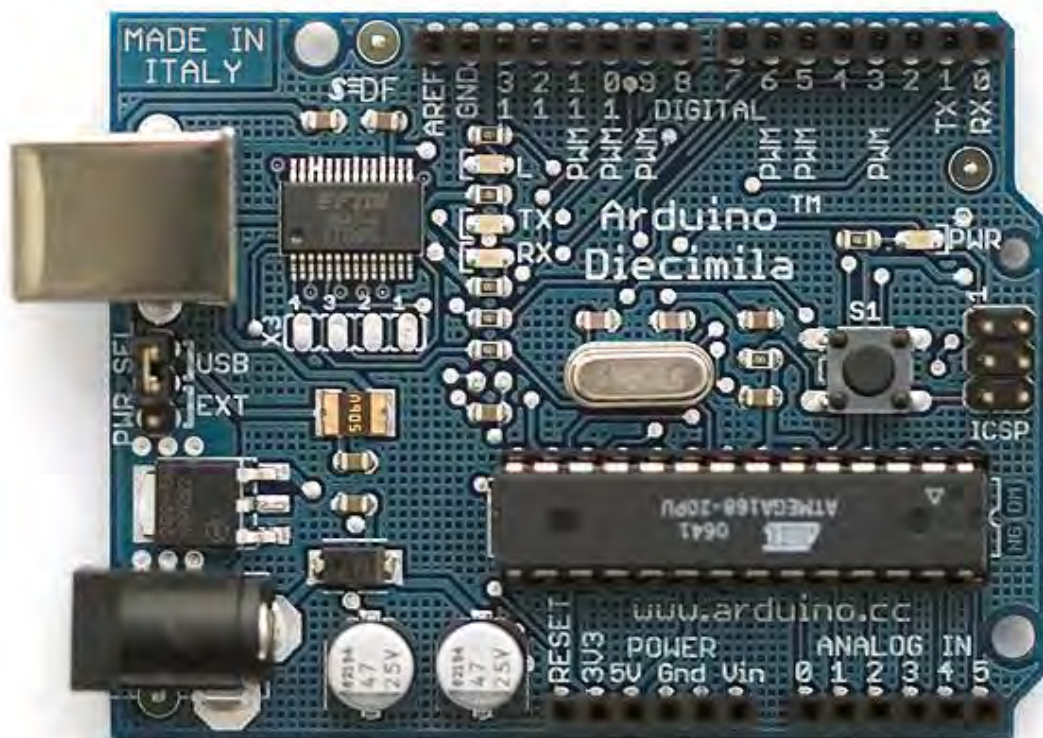


図 2.3 Arduino 制御ボード

Arduino ボードを制御するソフトウェアは、統合開発環境 (IDE) で開発する。IDE は、Windows、Macintosh、Linux で動作するマルチプラットフォームの Java アプリケーションであり、オープンソースで入手することができる<sup>3)</sup>。



図 2.4 Arduino IDE

### 第3節 制御方法

本研究では、自動灌水器を試作するにあたり、市販の自動灌水器（株式会社タカギ製 水分センサー付かんたん水やりタイマーセット G216）を改良して製作した。図 2.5 は試作した制御基盤の回路図であり、図 2.6 は実際の制御基盤である。

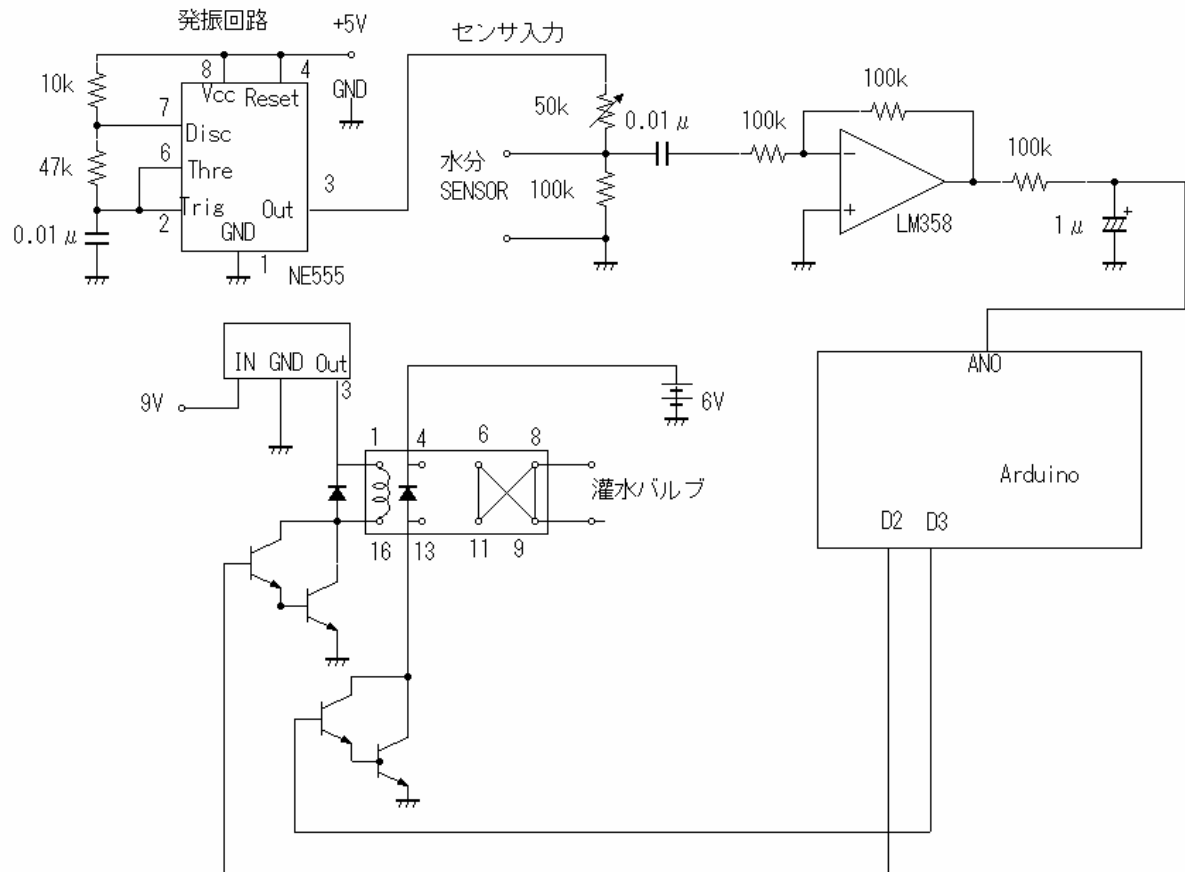


図 2.5 制御回路

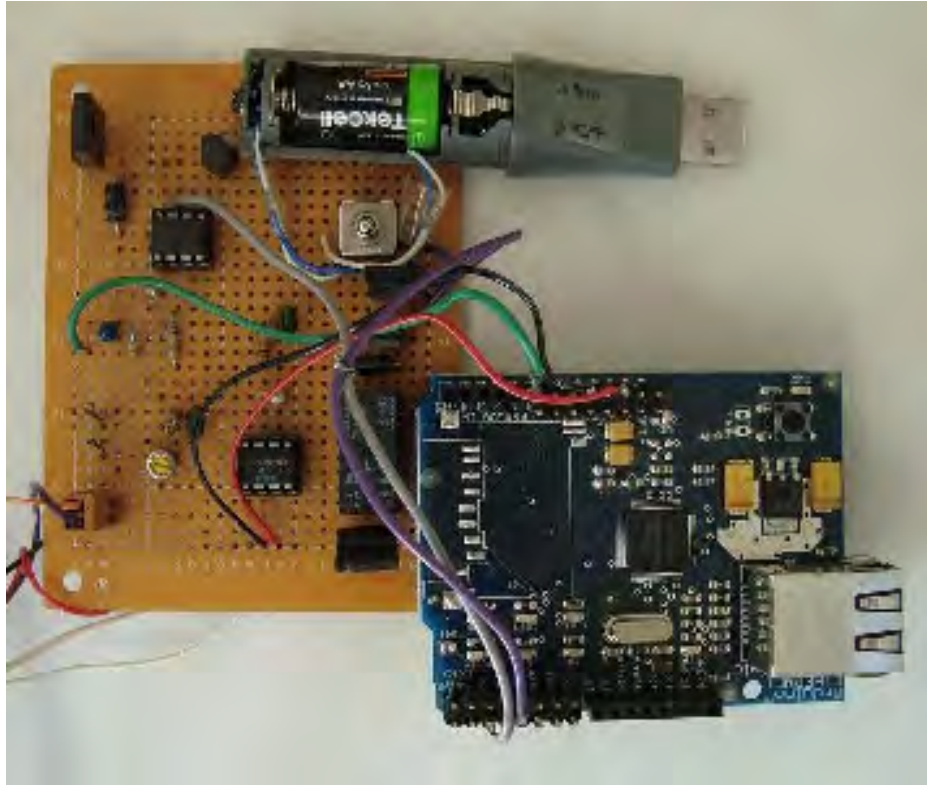


図 2.6 試作した制御回路

水分センサは、G216 の灌水器に付属するステンレス製の 2 本の電極を用いた（図 2.7）。このセンサを土壌中に埋め込み、電極間の電圧変化から土壌の水分状態を計測した。

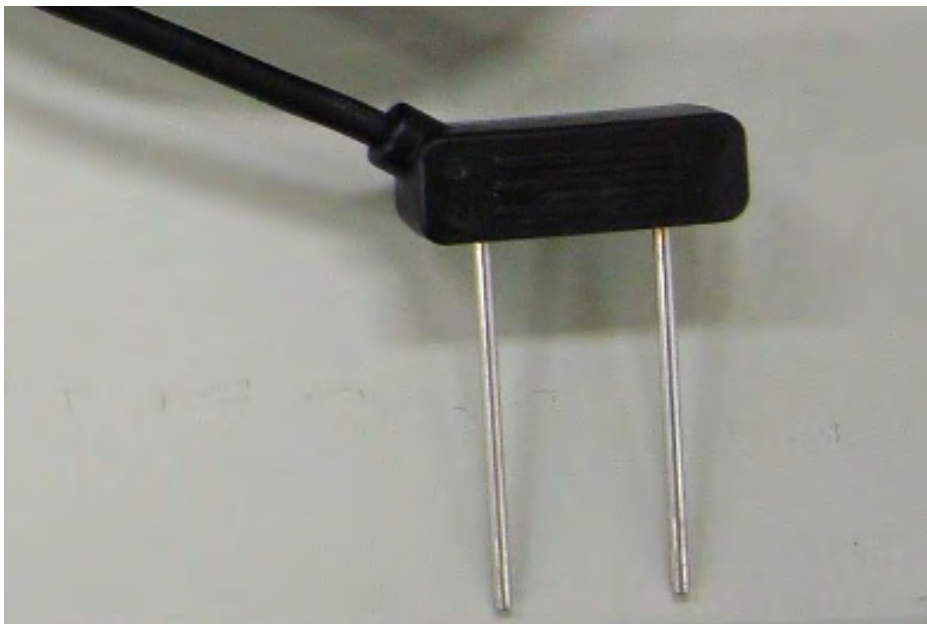


図 2.7 使用した水分センサ

電気分解を防ぐためにセンサには交流電圧を加え、電極用のインピーダンスを測り、センサに印加する電圧は 5V の電圧をタイマー555 にて 200Hz でパルス状（デューティ比 1）に発振し、LM358 の積分回路で Arduino のアナログ入力 0 番ピン（AN0）に入力した。そして pF 値に換算して設定値になるように、Arduino のデジタルポート 2、3 番（D2、D3）からデジタル値を出力し、リレーを制御した。



使用した G216 の電磁弁（図 2.8）の構造は、消費電力を抑えるため、信号が送られて弁が動作した後、その信号が遮断されても動作した状態を保持するように設計されていた。したがって、極性を反転させるため、リレーを用いて矩形波を2つのポートに与えることにした（図 2.9）。なお、制御回路の電源は 5V、電磁弁の制御用電源には単三電池 4 本による 6V を使用した。

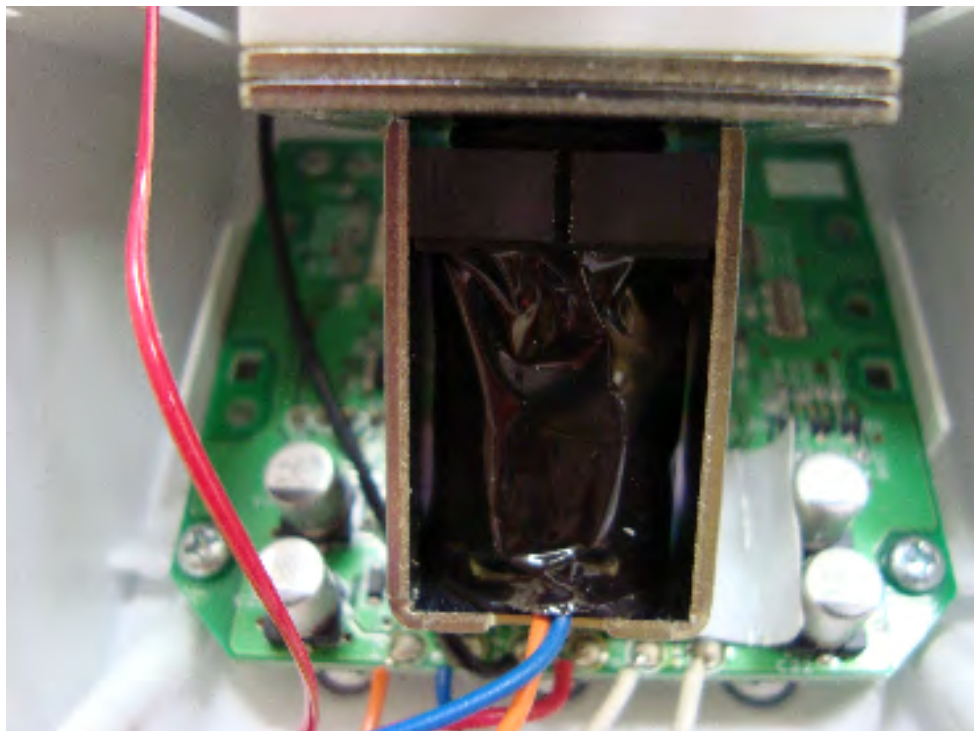


図 2.8 使用した電磁弁

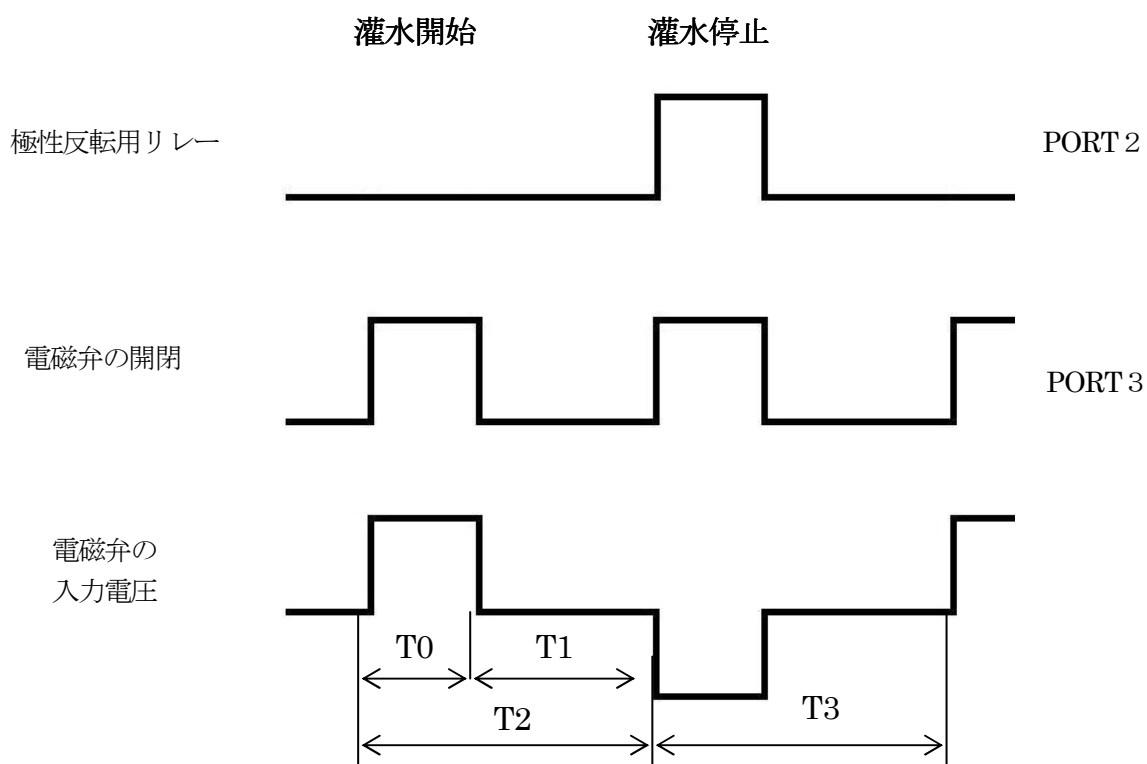


図 2.9 電磁弁のタイミングチャート

図 2.9 において、T0 は電磁弁への通電時間、T1 は待ち時間、T2 は灌水時間、T3 は灌水停止時間を表す。ここでは T0 を 500ms、T1 を 4500ms とし、T2 を 5000ms となるように設定した。T3 も同様の方法で 5000ms に設定した。これらは、センサの設置する深さ、圃場の広さなどに応じて任意に設定することができる。

灌水開始時期および灌水停止時期を設定するにあたり、前出の pF の測定値とセンサ部からの出力電圧 AN0 の関係を測定したところ、表 2.3 に示す結果が得られた。測定は、pF メータの検出部とセンサの埋没深さをともに 10cm とし、土壌の乾燥過程で行った。

表 2.3 pF 値と出力電圧の関係

pF	1.7	1.9	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8
出力電圧 (×5mv)	19	24	27	29	31	44	46	50	52	59

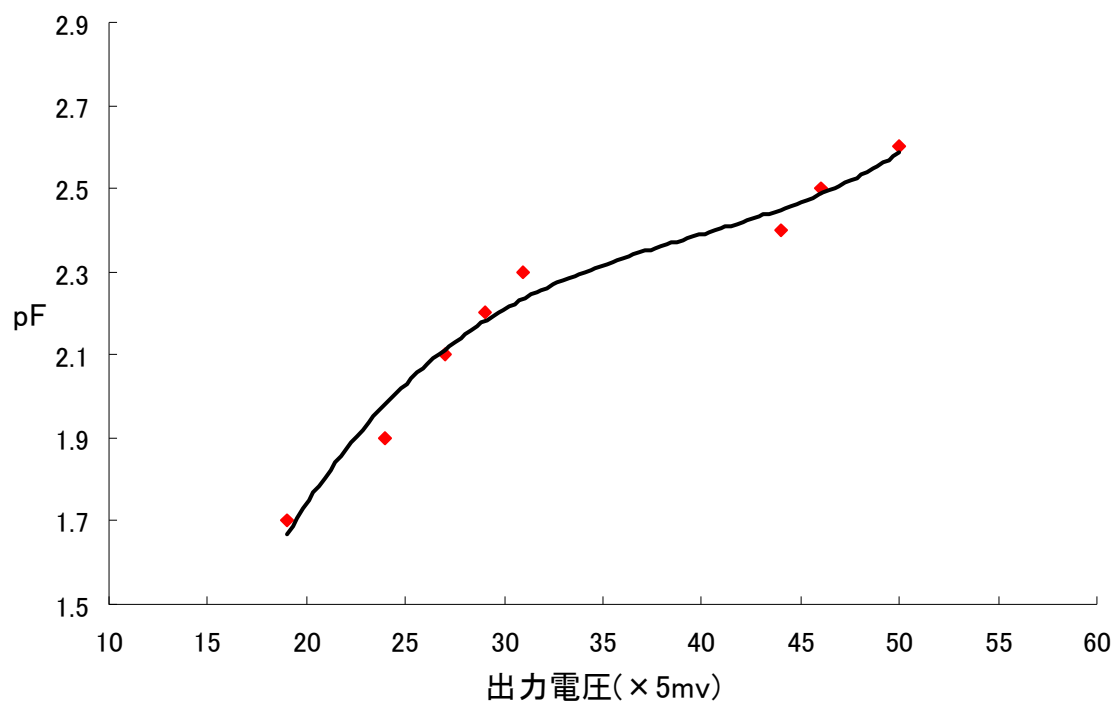


図 2.10 pF 値・電圧線図





図 2.11 pF 値と出力電圧の測定の様子

この結果から近似曲線（式 2.3）を求め、それをもとに灌水開始時期および灌水停止時期で設定した。なお、土壌の有効水分範囲は生育する植物や生育時期によって異なるが、本研究では一般的な範囲とされる pF1.7 から 2.3 を有効水分範囲としてプログラムを設計した。

$$y = 5 \cdot 10^{-5} x^3 - 6 \cdot 10^{-3} x^2 + 0.2493x - 1.2616 \quad (\text{式 2.3})$$

y : pF

x : 出力電圧（×5mv）

自動灌水システムの Arduino プログラムを付録資料 A3 に示した。

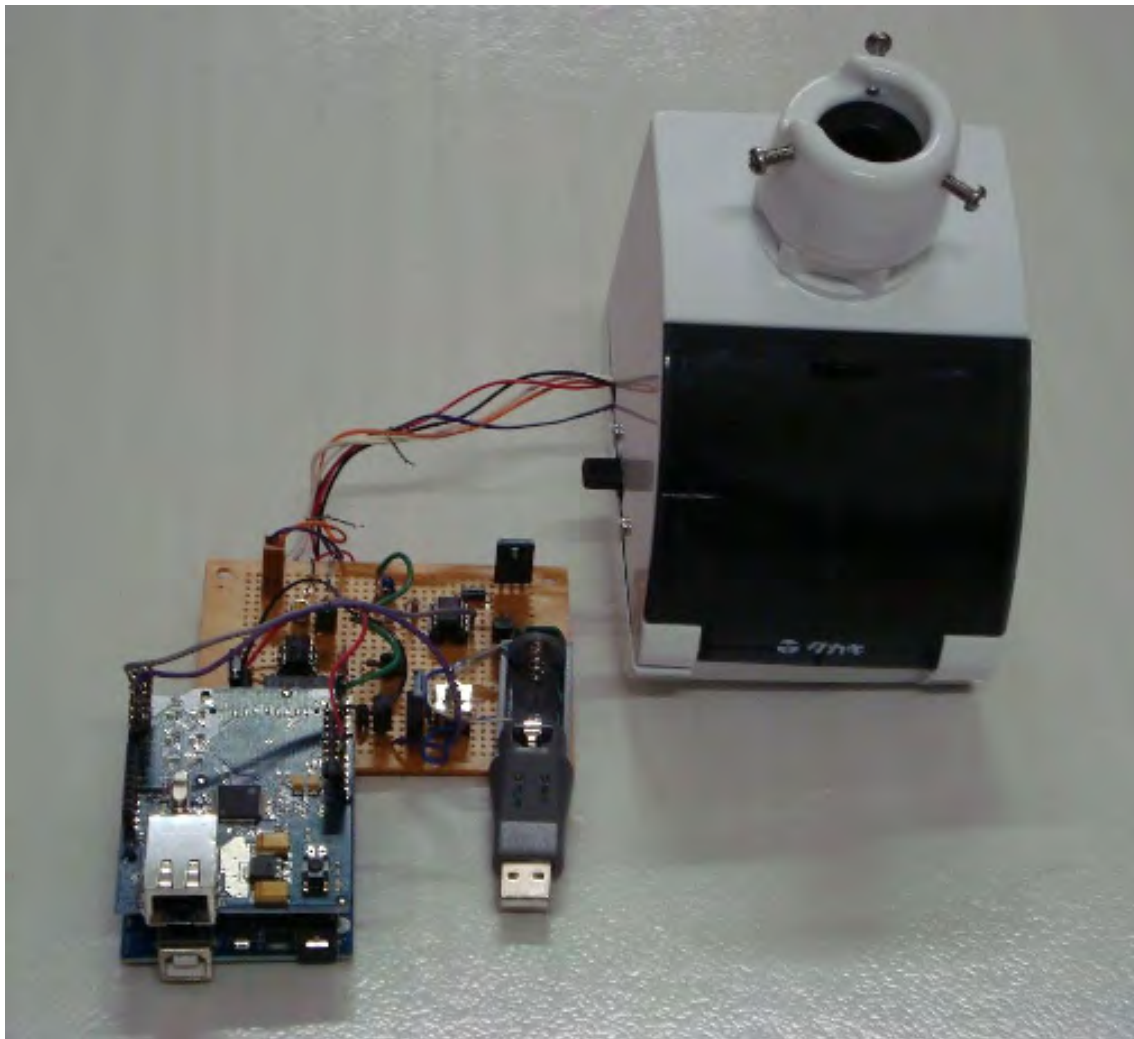


図 2.12 試作した自動灌水器

## 第3章 リモート灌水制御

### 第1節 システム概要

土壌の灌水状態に応じて自動で灌水を行う方法のほかに、ネットワークを介してリモートで灌水する方法も作業を省力化させる有効な手段であると考えられる。このことから、リモートコントロールによって灌水を行うシステムの試作を行った。図 3.1 にシステムの概要を示す。

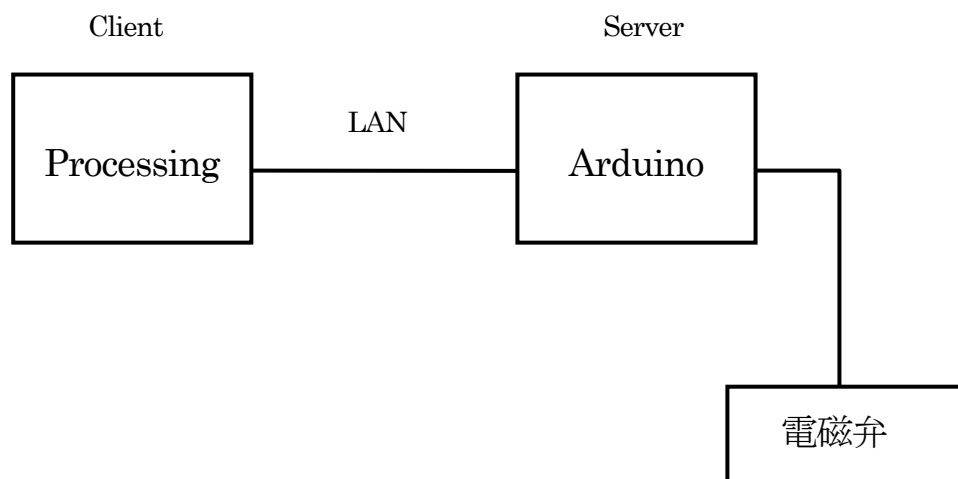


図 3.1 リモート灌水システムの概要

システムを構築するにあたり、クライアントコンピュータとサーバコンピュータを LAN で接続し、サーバコンピュータと Arduino をシリアル通信で結んで電磁弁の開閉を制御する方法<sup>1)</sup>があるが、より簡易に導入するために図 3.1 に示す制御系とした。図 3.1 において、Arduino をサーバとして電磁弁に接続し、後述する Processing をクライアントとし、両者を LAN で結び、ネットワーク越しに電磁弁の開閉を制御する。なお、ここではセンサを使用して水分状態を把握することは行っていない。

## 第2節 Arduino Ethernet Shield

Arduino をサーバとして機能させ、クライアントとイーサネット通信を行うために使用されるのが Arduino Ethernet Shield というオプションボードである。Arduino ボードの上に積み重なるように接続して使用される (図 3.2)。Ethernet Shield の基板には WIZnet 社の TCP/IP 処理チップが搭載されており、Arduino の AVR マイコン(ATmega328 あるいは ATmega168)とは SPI で接続される。ピン配置は Arduino と同じであるが、デジタル I/O ピンの 10 番から 13 番まで SPI 通信に使用されるため、センサなどを接続することはできない。クライアントコンピュータとのイーサネット通信を行うには MAC アドレスと IP アドレスを設定する必要がある。

MAC アドレスはイーサネット対応のハードウェアを一意に識別するアドレスである。Ethernet Shield の MAC アドレスは、株式会社スイッチサイエンスから購入した場合は、ボードの裏面に予め記載されてある 00-50-C2-97-20-11 のような 16 進数の 6 つの数値となる。記載がない、すなわち Ethernet Shield の MAC アドレスが不明の場合は、重複しないように任意で設定する。また、Ethernet Shield には MAC アドレスの記憶領域がないため、プログラムの中で明示的に設定する必要がある。

IP アドレスは、ネットワークに接続されたコンピュータや通信機器に割り振られた識別番号である。したがって、Ethernet Shield には他のコンピュータや通信機器と重複しない範囲で任意で設定する。今回の実験には、技術科に割り振られた 160.29.112 サブネットの中の IP アドレスを使用した。



図 3.2 Arduino ボードとその上にスタックされた Arduino Ethernet Shield

電磁弁のコントロールはデジタル入出力ピン 2、3 番を使用した。

既述の通り、省電力型の電磁弁であるため、図 3.5 のタイミングチャートのようにポート 2 をリレー入力とし極性を反転させるため使用し、ポート 3 を電磁弁本体の駆動信号とした。

### 第3節 Processing

Processing は、MIT（米国マサチューセッツ工科大学）メディアラボで Ben Fry と Casey Reas らによって開発されたプログラミング環境および API（Application Program Interface）である<sup>2)</sup>。開発当初は、画像処理やアニメーションといった芸術家およびデザイナーをターゲットとした特定分野向けの位置づけであったが、現在では建築、人文科学、統計学など利用範囲が広がってきている。

Processing は、

- ・エディタおよび実行コンソール画面を持つ IDE（統合開発環境）が付属
- ・マルチプラットフォーム
- ・オープンソースプロジェクト
- ・Java をベースに単純化したプログラミング言語
- ・プログラムコードの記述が容易
- ・Arduino との相性が良い<sup>3)</sup>
- ・プログラミング初心者にとって扱いやすい環境
- ・画像処理が可能
- ・シリアル通信機能を有する
- ・ネットワーク通信機能を有する
- ・サーバクライアントシステムの構築が可能
- ・Applet として web で公開できる

などの特徴を有していることから、教育利用に適したプログラム開発環境と考えられる。

以上のことから、本研究ではクライアントに Processing を使用することとした。

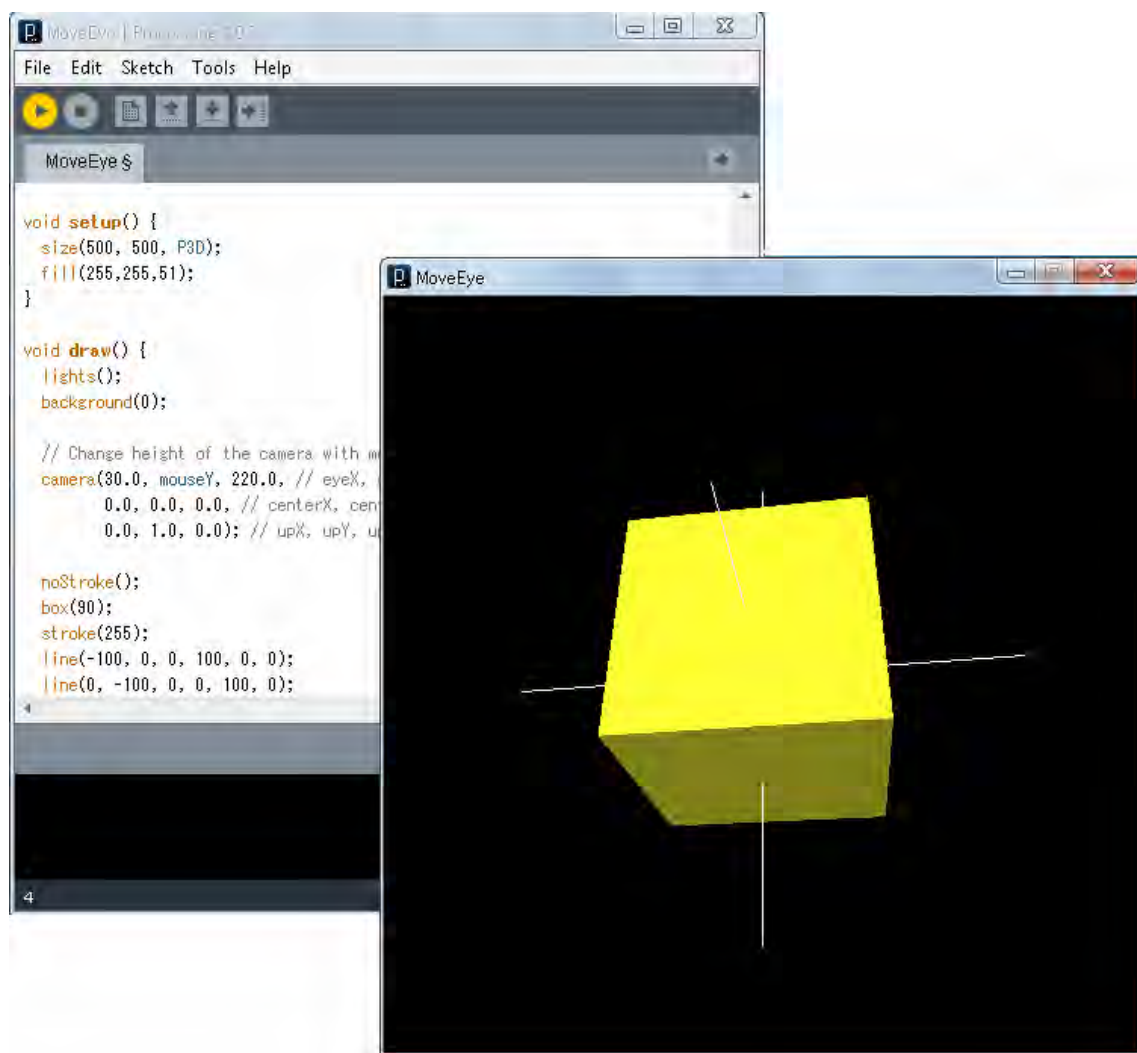


図 3.3 Processing の画面構成 (3D 仮想物体)

## 第4節 制御方法

プログラムを作成するに際し、**Processing** ではネットワークライブラリを、**Arduino** ではイーサネットライブラリをそれぞれインポートすることによってサーバー・クライアント間のネットワーク通信が行えるようになる。

図3.4は、クライアントである **Processing** の実行画面である。緑のボックスが電磁弁を開くボタンで、赤のボックスが電磁弁を閉じるボタンであり、それぞれの色の明暗によって電磁弁の動作状態を区別している。

これらのボタンを作成するにあたり、**Processing** には GUI (Graphical User Interface)コンポーネントが提供されていないため、アプリケーション開発用ソフトウェアのようにフォームにボタンを配置してクリックされた後の処理をコーディングすることは行われない。かわりに下記のコマンドを用いてマウス操作に応じた処理を記述することで、ボタンの代用を作成することができる。

<code>mousePressed()</code>	マウスを押下時にプログラムを処理する。
<code>mouseReleased()</code>	マウスを押下時にマウスを離すとプログラムを処理する。
<code>mouseMoved()</code>	マウスを動かしている間にプログラムを処理する。
<code>mouseDragged()</code>	マウスをドラッグした時にプログラムを処理する。
<code>mouseX</code>	マウスの X 座標。
<code>mouseY</code>	マウスの Y 座標

これらの中から、`mousePressed()`、`mouseX`、`mouseY` の3つのコマンドを使用し、マウスポインタがボックス内部にある場合にクリックすると、サーバに信号が送信され、**Arduino** で受信し、電磁弁の開閉を制御するようにプログラムを設計した。

また、図 3.5 に示すように、電磁弁およびリレーの動作タイミングは図 2.6 と同様である。但し、ここでは **T1** と **T2** を設定せずに、灌水時間をクライアントによって自由に決定できるようにした。

付録資料 A4 にリモート灌水制御のクライアントプログラム、付録資料 A5 に同サーバプログラムを示した。

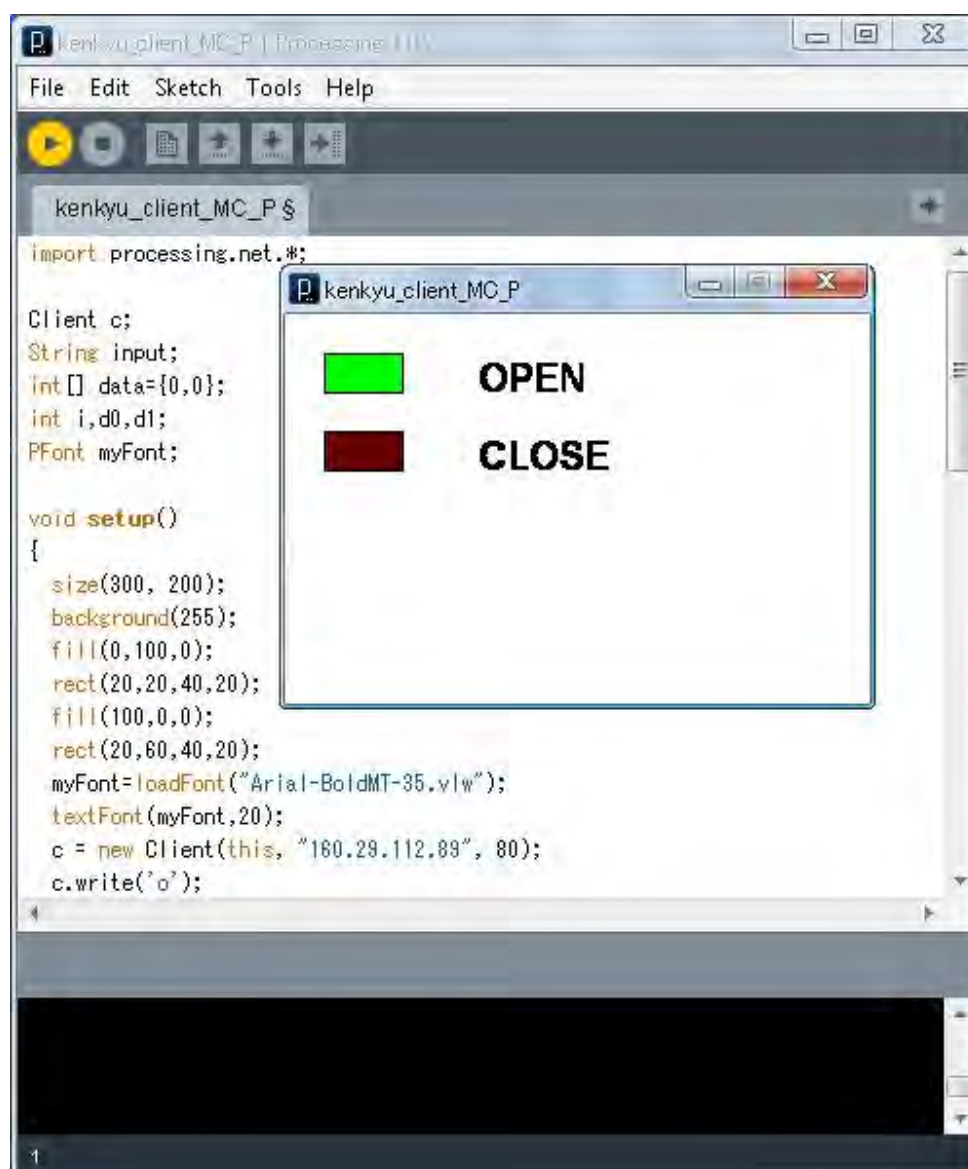


図 3.4 Processing の実行画面（灌水開始時）



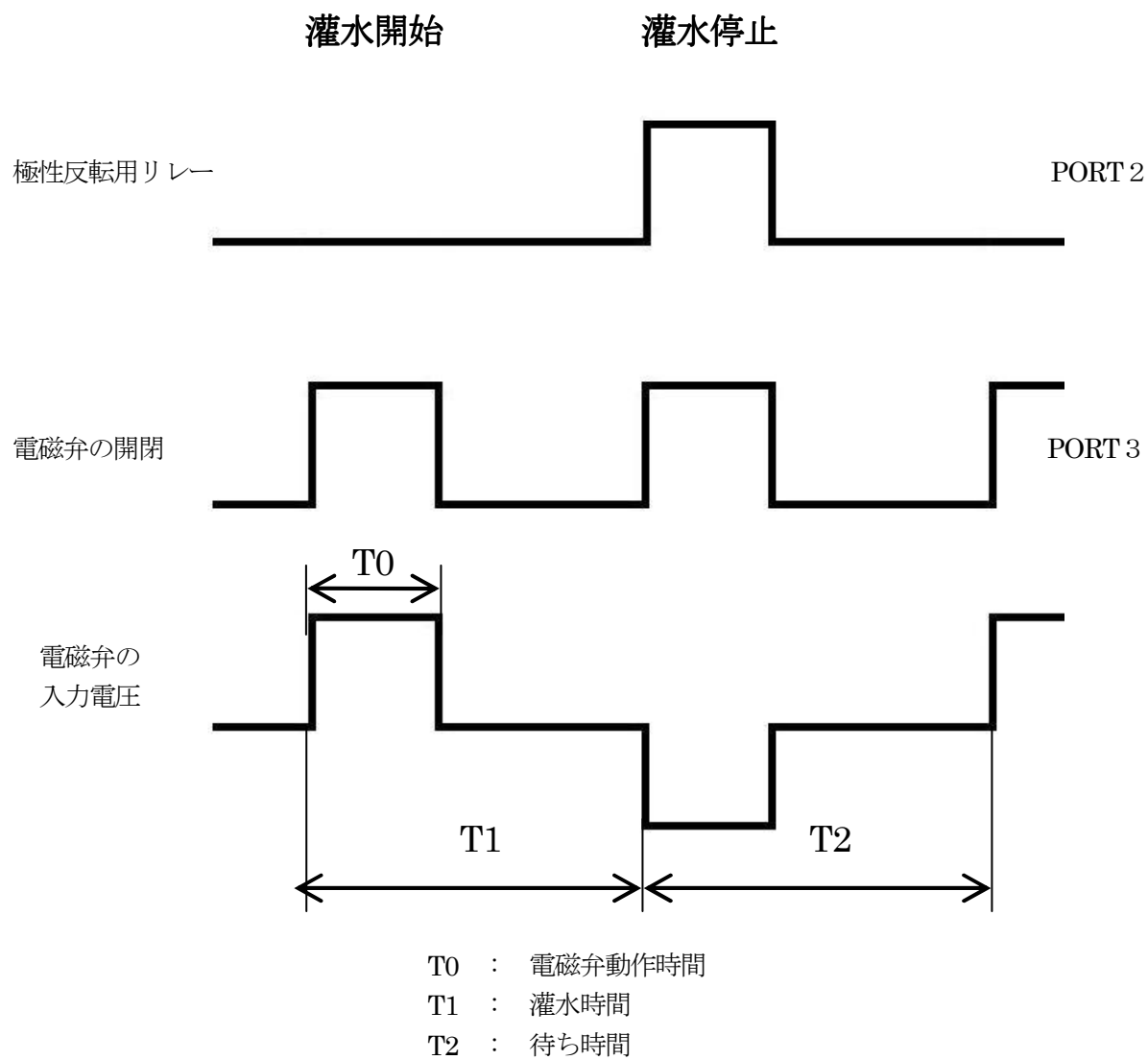


図 3.5 リモート灌水のタイミングチャート

## 第5節 実行結果

**Processing** 側でイベント処理を行わない場合に、フレームレートを大きく設定すると、常にポート 2、3 の状況が送信され、ネットワークは輻輳する。

例えば、フレームレートを 10 に設定すると受信タイミングを失ったパケットが受信バッファに蓄積され、速度が遅くなった。

このような状況を避けるため、ボタンが押された時のみネットワークに送信するイベント処理を記述した。その結果、ほぼリアルタイムに電磁弁をコントロールすることが可能となった。

## 第4章 ネットワーク計測

### 第1節 システム概要

次に、ネットワークを介して作物の生育にかかわる環境の諸条件の計測を試みた。ここでは、土壌の水分状態（pF）、気温、湿度の3つを計測対象とし、それらをリモート灌水制御と同様に、**Arduino** をサーバとして機能させ、**Processing** をクライアントとした簡易計測システムの設計および構築を行った。システムの概要を図 4.1 に示す。

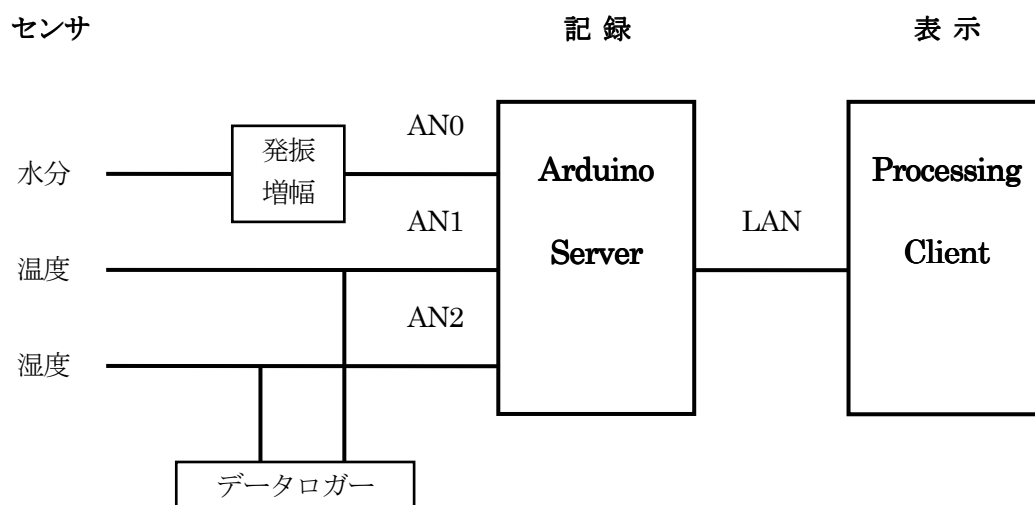


図 4.1 ネットワークを利用した簡易計測システムのブロック図

IP アドレスは、リモート灌水制御の際に用いた 160.29.112 サブネットを使用した。また、pF の計測にはこれまでと同様に製作した水分センサーを使用して行い、気温と湿度も **Arduino** で計測するとともに USB データロガーを用いて計測した。データロガーは、LASCAR electronics 社の EL-USB-2 を使用した（図 4.2）。



図 4.2 気温、湿度の計測に使用したデータロガー

表 4.1 データロガーの仕様

相対湿度	記録数	16382
	測定範囲	0~100%RH
	短期再現性	±0.1%RH
	精度	±3.0%RH (20~80%RH における総合誤差)
	内部分解能	0.5%RH
	長期安定性	0.5%RH
温度	記録数	16382
	測定範囲	-35°C~+80°C (-31 ° F~+176 ° F)
	再現性	±0.1°C (±0.2 ° F)
	精度	±0.5°C (±1 ° F) (20~80%RH における総合誤差)
	内部分解能	0.5°C (1 ° F)
露点	精度	±1.1°C(±2 ° F)(25°C、40-100%RH における総合誤差)
精度 (Overall Error)		±1°C (±2 ° F)
ログ間隔		10 秒ごと~12 時間ごと
動作温度		-35°C~+80°C (-31 ° F~+176 ° F)
1/2AA 3.6V リチウム電池		約 1 年 (ログ間隔や気温に影響する)

本実験で使用したデータロガーは、サーミスタと湿度センサによって温度や湿度変化を電気抵抗の変化に変換している。このため、新たに Arduino ボードのアナログ入力ポートの 1 番と 2 番を使用して (図 4.3)、Processing で結果表示をすることとした。なお、本実験では電磁弁の開閉の制御は行わないため、図 2.5 と異なり、電磁弁駆動用の 6V 電源は使用せず、センサ駆動回路用の 9V 電源のみを使用した。

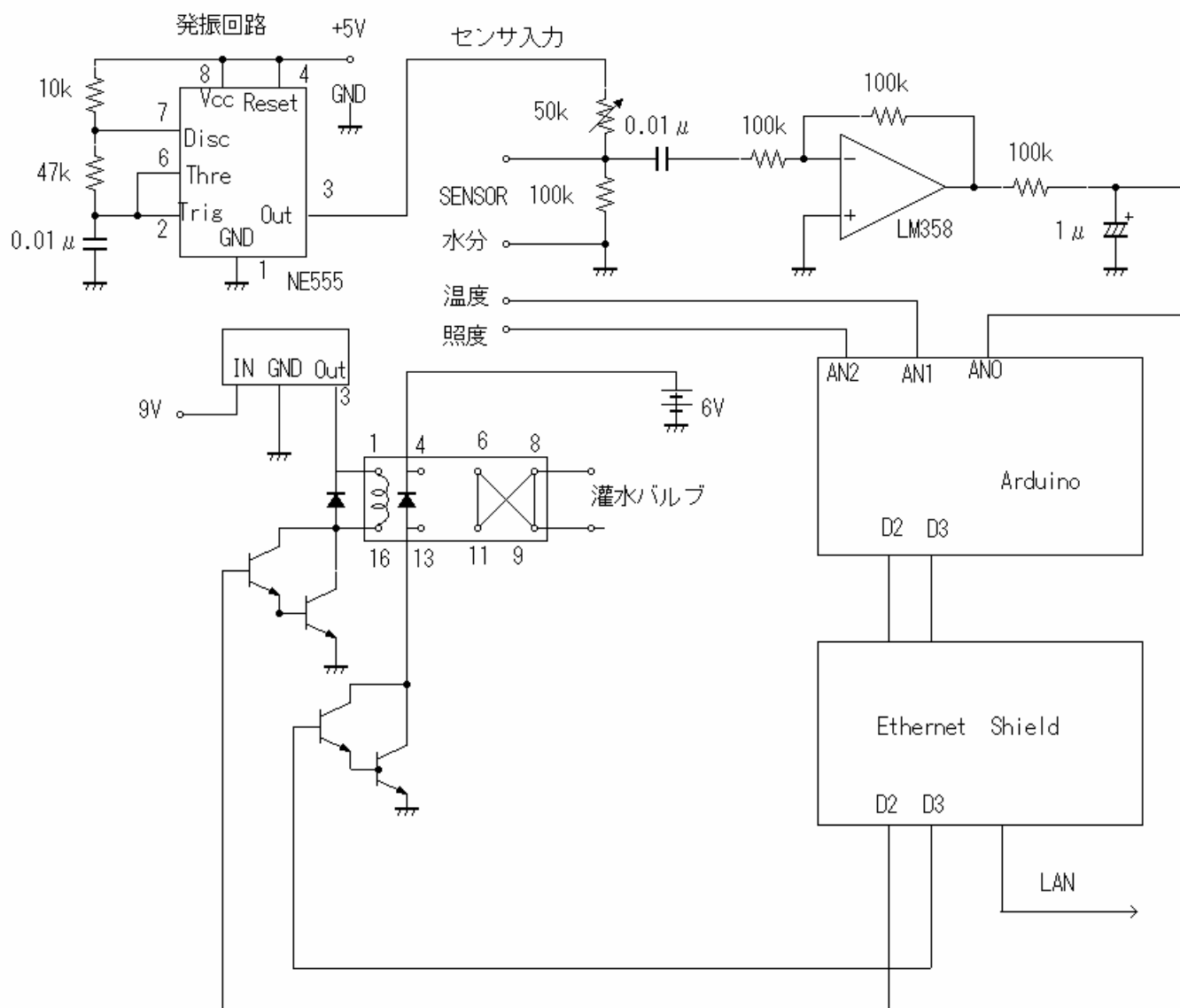


図 4.3 温度・湿度センサを付加させた回路図

## 第2節 計測方法

図 4.4 は、Arduino サーバから計測した pF 値の変化をクライアントの Processing に表示させたものである。

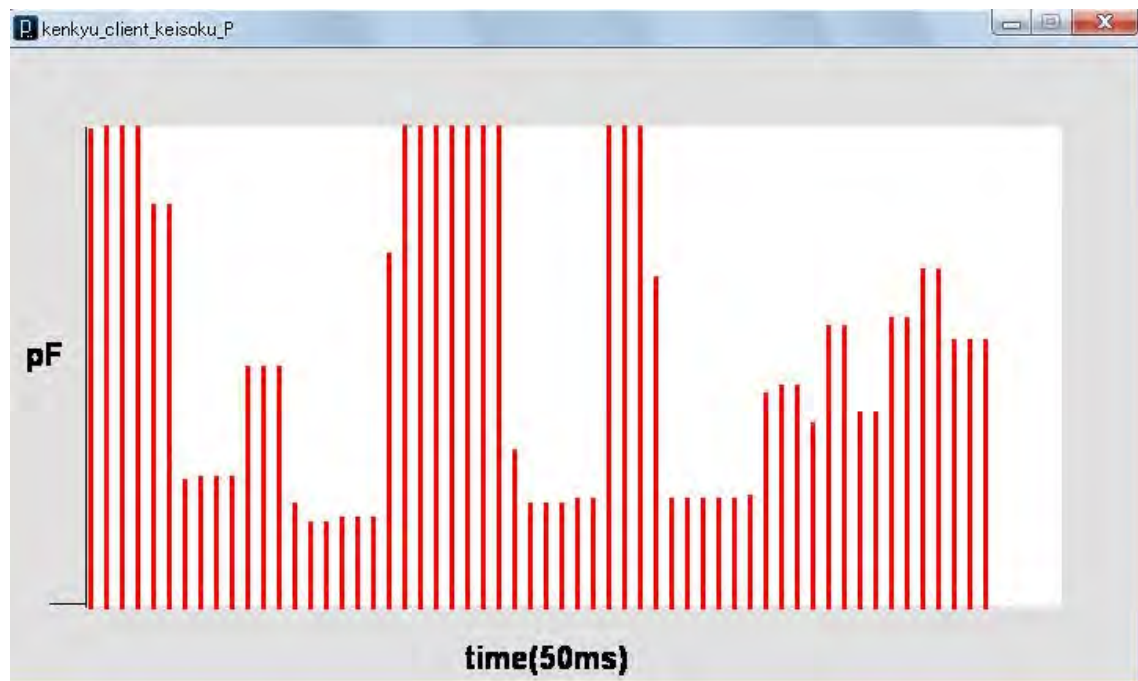


図 4.4 pF 値の計測結果のグラフ表示

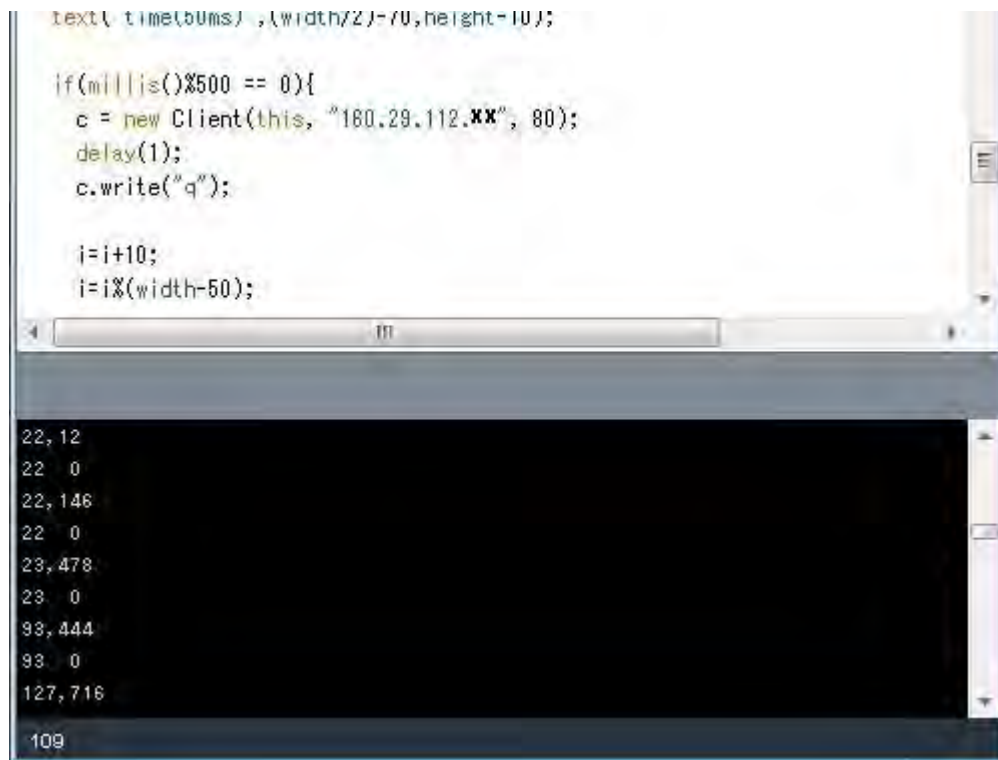
図 4.4 は、動作確認のために Arduino サーバに接続した土壤水分センサを手動で湿った布で触れたり外したりした時の pF 値の変化を棒グラフで時系列表示したものである。Arduino から受信したアナログ値を Processing の map() メソッドを用いて値の範囲を置換・増幅して表示させた。また、表示間隔は、50ms に設定した。pF 値が大きいほど土壤が乾燥状態であることを示し、小さいほど土壤が湿潤状態であることを示している。変化量を確認するために濡れた布をセンサにあてて計測したため著しい変化を示しているが、実際の土壤ではこのような短時間での著しい変化は起こりえない。これより、土壤の水分状態のリモート計測が可能であることを確認することができた。同様に、他の生育環境も該当するセンサを用いればリモート計測を行うことができる。

リモート計測システムのクライアントプログラムを付録 A6 に、サーバプログラムを付録 A7 に示した。

### 第3節 計測結果

pF や気温、湿度をネットワークを介して計測することを試みた結果、Processing で計測値をリアルタイム表示できることが確認された。しかし実際には、図 4.5 に示すように明らかに不正確なデータも混在して受信されていて、グラフ表示の際にも正確に 50ms 間隔では表示されず、厳密にはリアルタイムでのデータの送受信が行われていないことが判明した。

この問題を解決する方法として、フレームレートの調整とパケットフィルタリングで特定のサーバからのデータのみ受け付けるようにすること、および各種の計測結果を受信および表示を行う Processing 側のプログラムの中に新たに補間プログラムを挿入することが考えられる。



```
text(t, time(bums), (width/2)-70, height-10);

if(millis()%500 == 0){
  c = new Client(this, "180.29.112.xx", 80);
  delay(1);
  c.write("q");

  i=i+10;
  i=i%(width-50);
}
```

22, 12  
22, 0  
22, 146  
22, 0  
23, 478  
23, 0  
93, 444  
93, 0  
127, 716  
109

図 4.5 Processing の受信データ（カンマを挟んで左が pF、右が気温データ）

このうちフレームレートについては現在の設定値を 10fps を 1ps、表示間隔を 50ms から 1sec とすることでほぼ不正データの混入を避けることができた。

実際の運用では表示間隔は少なくとも 10 分間隔などが予想されるため、不正データの混入は起こりにくいと考えられるが、安全のため測定値の前後 3 点をとって異常値を削除するなどの対策が考えられる。

## 第5章 画像解析を用いた土壌酸度の測定

### 第1節 実験の目的

土壌の酸度は作物の生育に強い影響力をもち、酸性土壌もアルカリ性土壌も培地としては不適當であり、例外的に酸性を好む作物を除けば、中性や微酸性の土壌が好まれる。日本の土壌の多くは酸性土壌であるため、農耕地では炭酸カルシウム、生石灰、消石灰などの土壌改良資材を施用することにより、作物の生育に適した土壌酸度に調整される。したがって、土壌酸度を測定することは作物を栽培するうえできわめて重要な要素であり、栽培学習を行う前には測定しておくことが望ましい。

土壌の酸性あるいはアルカリ性は、一般にその pH 値を指標として判断され、久馬らによると『大部分の無機質土壌の pH3.5 ないし 10 の範囲にあり、そのなかでも 4 ないし 8.5 の範囲のものが多い』とされている<sup>1)</sup>。

土壌酸度は、市販の pH 測定用試験紙やデジタル pH メータを使用することにより、簡易に測定することができる。しかしながら、pH 試験紙のように試験紙の色を比色表と比較して測定する方法は、測定者の視覚という感覚量が判断基準となり、定量的に評価することが難しく、とくに色覚異常者にとっては大きな弊害となることが予想される。また、シェファー、シャハトシャーベルらによると、『圃場では大まかな調査を行なうために pH 測定用試験紙が用いられるが、その pH 値はきわめて不正確である ( $\pm 0.5\text{pH}$ )。』とされている<sup>2)</sup>。デジタル pH メータを使用することによって定量的な計測は行えるが、高価であるため、汎用の教育利用という立場では比色表を用いた画像計測による測定が有効な手段であると考えられる。

比色表を用いた画像計測には、

- ・人間の感覚量に依存せず、判定・評価を定量化することができる
- ・デジカメで撮影した比色表から正確な pH がわかる
- ・教育利用において、コンピュータの色の表現方法 (RGB 値) の教材となる

などの利点を有している。

したがって、本実験では比色表を用いた画像計測によって土壌の酸度測定を行い、その精度について他の手法との比較・検討を行うことにした。



## 第2節 ImageJ

本実験では、画像処理ソフトウェアに **ImageJ** を用いた。**ImageJ** は、アメリカ国立衛生研究所 (NIH) が開発したオープンソースの画像処理ソフトウェアである。前身は **NIH Image** で、主に医療・生物分野の研究用として利用されてきた。

**ImageJ** は、**Java1.1** 以降の **Java** 仮想マシン (VM) を搭載しているコンピュータ上で動作し、**Windows**、**Mac OS X**、**Linux** でダウンロード及びインストールが可能である。**TIFF**、**BMP**、**JPEG**、**GIF**、**MPEG**、**AVI** などの多くの画像ファイルフォーマットに対応しており、**8 bit**、**16bit**、**32bit** 画像の表示、加工、処理、分析など自由度の高い画像処理が行えるため、ピクセル値から面積、距離や角度、密度、成分濃度などを計算処理することができるなど、機能が充実している。

このような特徴を有する **ImageJ** は、従来の医療・生物分野での利用にとどまらず、環境計測や物体の力学運動の解析なども高い精度で行えることが報告されており、その応用範囲は広いと言える<sup>3) 4)</sup>。

なお、**ImageJ** は、公式サイト<sup>4)</sup> よりダウンロードして入手することができ、最新版のバージョンは、2010 年 1 月末現在、**1.43n** である。

### 第3節 手順

屋内における土壌の pH の測定は、一般に、採取して風乾させた土壌試料に等量ないし 2.5 倍量の純水または塩含有の溶液を加え、1～3 時間後に測定する。

本実験では、一般的な測定法に則り、土壌試料 20g に対して純水 50ml を加え、およそ 3 時間後に pH を測定した。pH の測定は、

- 1) 市販の簡易測定液を用いた目視による測定
  - 2) デジタル pH 計による測定
  - 3) 簡易測定液の画像解析による測定
- の 3 通りの方法で行った。

なお、測定は同様の手法で 3 回行った。実験に使用した器具および試料は、表 5.1 に示すとおりである。

表 5.1 実験に使用した器具と試料

土壌試料	赤玉土、バーミキュライト、農場で採取した土壌 それぞれ 20g
懸濁液	純水 50ml
酸度測定液 試験管 比色表	・住友化学園芸株式会社製 「一目でわかる酸度 pH 測定液 アースチェック液」 ・測定液の成分：アルコール類 ・測定範囲：pH4.0～8.5 ・解析度：0.5pH
デジタル pH メータ	・佐賀エレクトロニクス株式会社製 「Waterproof pH pen」 ・測定範囲：pH0～14 ・解析度：0.1pH ・精度：±0.1pH
デジタルカメラ	・SONY 製 「Cyber-shot DSC-W110 / W120」 ・カメラ有効画像素：約 720 万画素
ソフトウェア	ImageJ、Microsoft Excel
その他	三角フラスコ、ビーカー、電子秤、薬さじ



図 5.1 使用したデジタル pH メータ

デジタル pH メータのセンサには、ガラス電極が使用されている。一般にガラス電極法と呼ばれ、ガラス電極と比較電極の間の電位差から pH を測定する方法である。ガラスの薄膜の内側と外側に pH が異なる溶液があると、薄膜部分に pH 差に比例した起電力  $V$  が発生する (式 5.1)。この薄膜はガラス電極膜と呼ばれ、プロトン伝導性をもつ。通常、ガラス電極の内部には中性溶液 (pH7.0 に調整された KCl 溶液) が用いられるため、電極膜に発生した起電力を測定することにより、被検液の pH を求められる。ガラス電極法は、古くから pH 測定の一般的な方法として用いられている。

$$V = \frac{RT}{F} \log \frac{C_{out}}{C_{in}} \quad (\text{式 5.1})$$

- $F$  : 1mol の電子の電気量 ファラデー定数
- $R$  : 気体定数
- $T$  : 絶対温度
- $C_{in}$  : 電気内  $[H^+]$  濃度 [mol/l]
- $C_{out}$  : 電気外  $[H^+]$  濃度 [mol/l]

1. 土壌試料 20g と純粋 50ml を三角フラスコ内に入れ、混合させた後、不純の混合を避けるため、アルミホイルでフラスコ上部に封をする。



図 5.2 測定する溶液の準備（左から土壌、赤玉土、バーミキュライト）

2. 3時間経過した後、溶液をろ過させて上澄み液を取り出す。

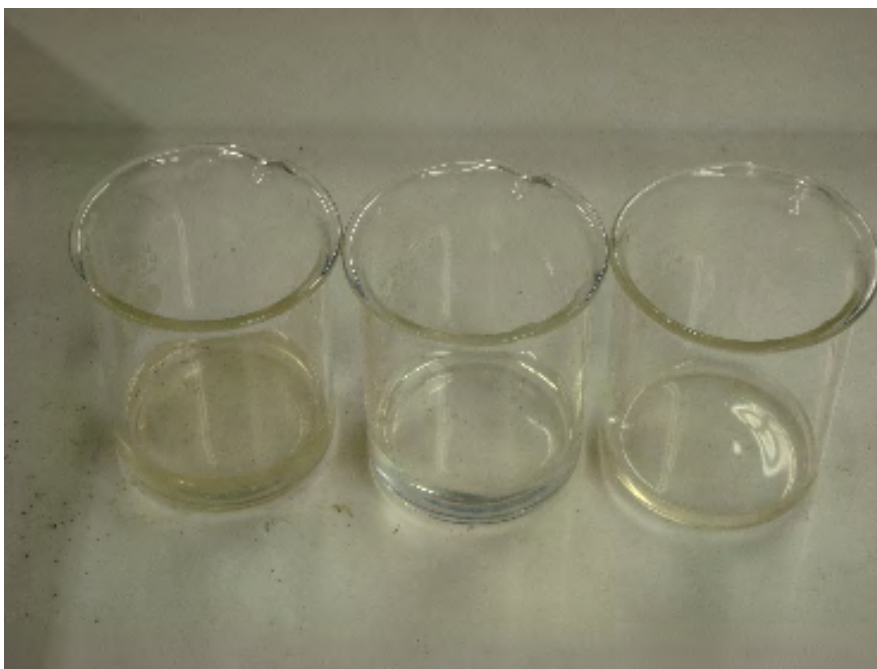


図 5.3 上澄み液（左から土壌、赤玉土、バーミキュライト）

3. 上澄み液にデジタル pH 計を入れ、pH を測定する。
4. 市販の酸度測定液に添付の専用試験管に上澄み液を 2.5cc 取り、測定液を 3 滴加え、よく振った後、比色表と比較して pH を測定する。測定した後、画像解析に使用するため、カメラで撮影を行う。

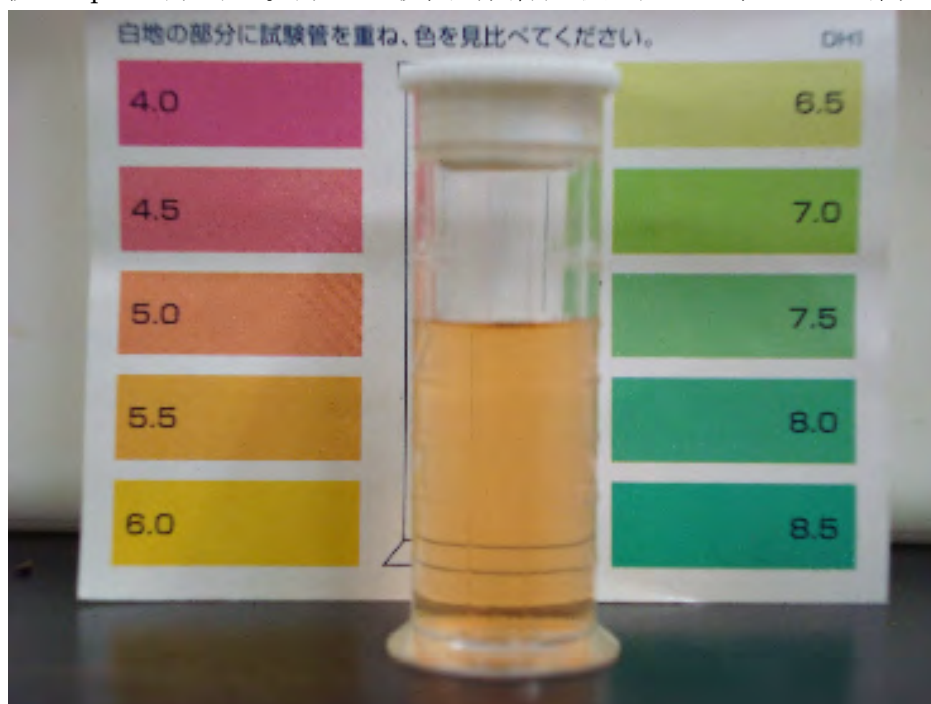


図 5.3 酸度測定液と比色表による測定（土壌）

5. 撮影した画像をコンピュータに取り込んだ後、ImageJ を起動させ、「File」→「Open」で ImageJ に画像を読み込む。（画像フォーマットは JPEG）

6. 「Image」→「Color」→「RGB Split」で8ビットのグレースケール画像のR(red)G(green) B(blue)の各成分に分解する。

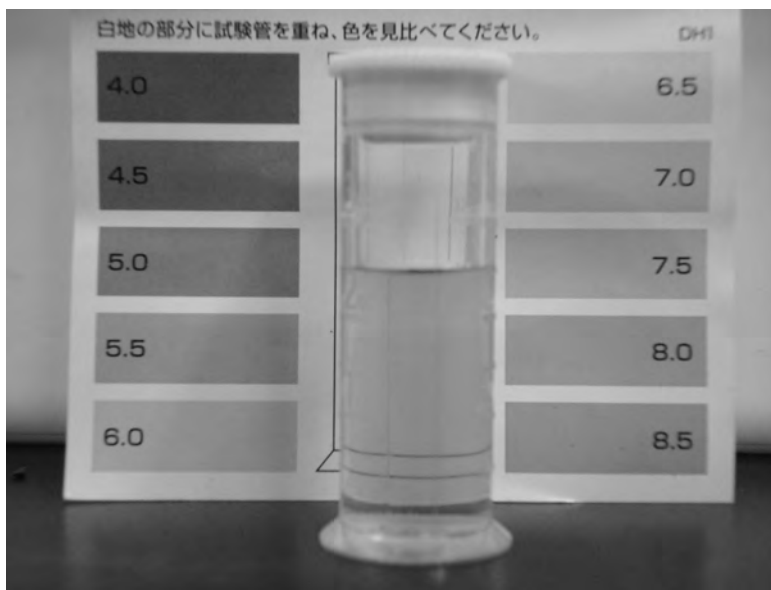


図 5.4 RGB 分解 (図は G 成分の画像)

7. 「Rectangular selections」ツールで計測する選択枠を決定する。選択枠の幅と高さを大きく取ると広い範囲の平均値を求めることができるが、出来るだけ影の部分を避けるため、本実験では選択枠の幅と高さをそれぞれ 60pixels とした。これは試験管の影が写真に入ることがあるため、影の影響のない部分を比較するために選んだサイズである。また、試料の色情報は、試験管の画像中の最も呈色が鮮やかな中央付近の部分をとった。

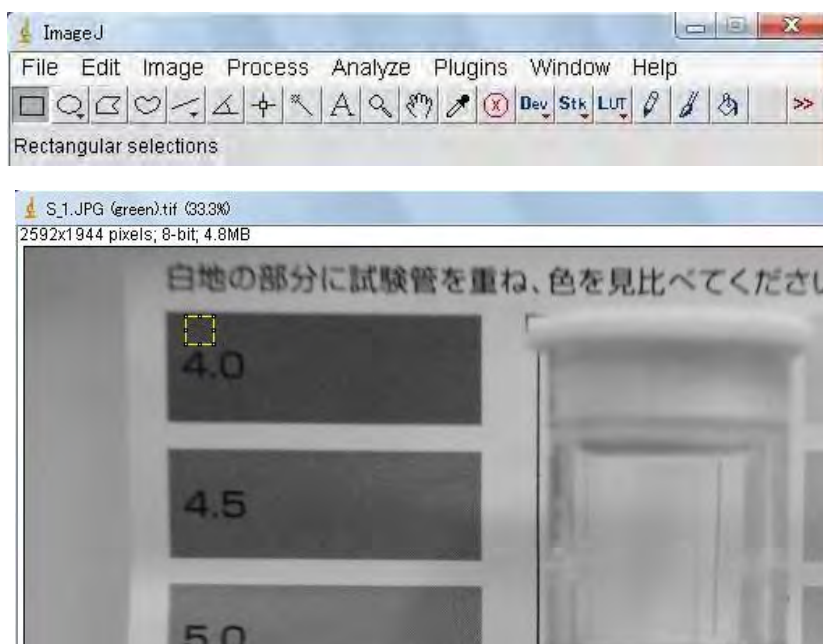
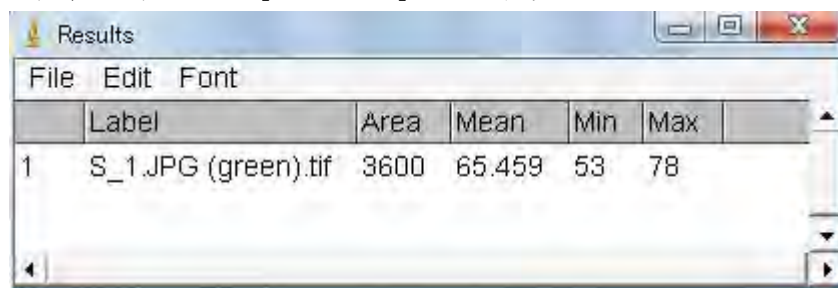


図 5.4 測定範囲の選択 (pH4.0 の色見本)

8. 「Analyze」→「Measure」で計測結果が Results ウィンドウとして新たに生成される。図 5.5 において、選択範囲の面積、色の最大値、最小値、平均値の計測結果が得られていることが分かる。なお、計測項目は、「Analyze」→

「Set Measurements」で変更、設定することができる。同様の操作を他の pH 値の色見本および試料で行い、Results ウィンドウの「File」→「Save」でタブ区切りのテキストデータとして保存する。



	Label	Area	Mean	Min	Max
1	S_1.JPG (green).tif	3600	65.459	53	78

図 5.5 計測結果 (pH4.0 の色見本)

9. テキストデータを Excel に取り込み、Excel 形式として保存する。

10. 同様の操作を RGB のそれぞれの成分の画像について行い、縦軸に pH、横軸に RGB のそれぞれの成分をとり、グラフ化する。

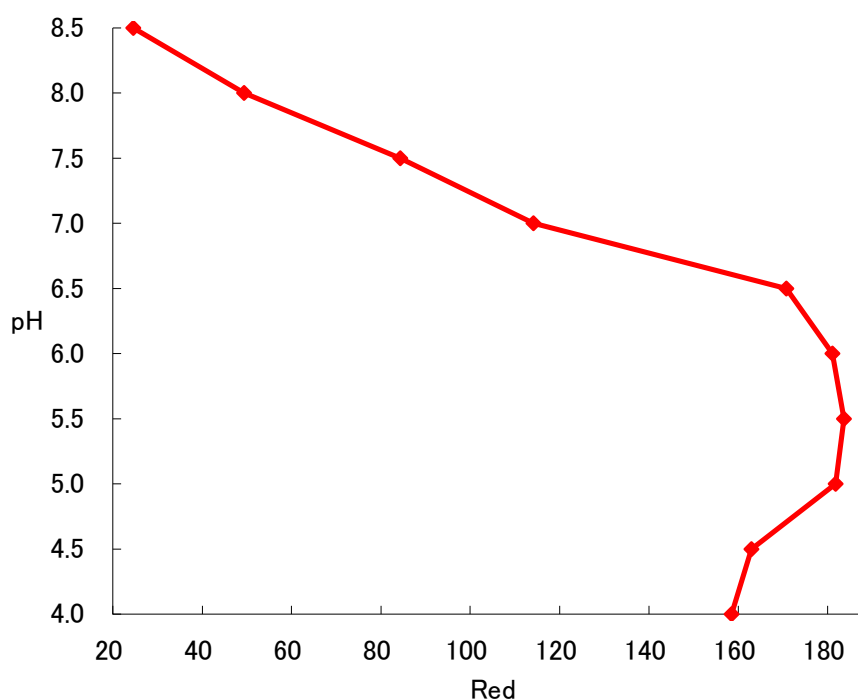


図 5.6 R-pH 線図 (土壌の 1 回目の実験結果)

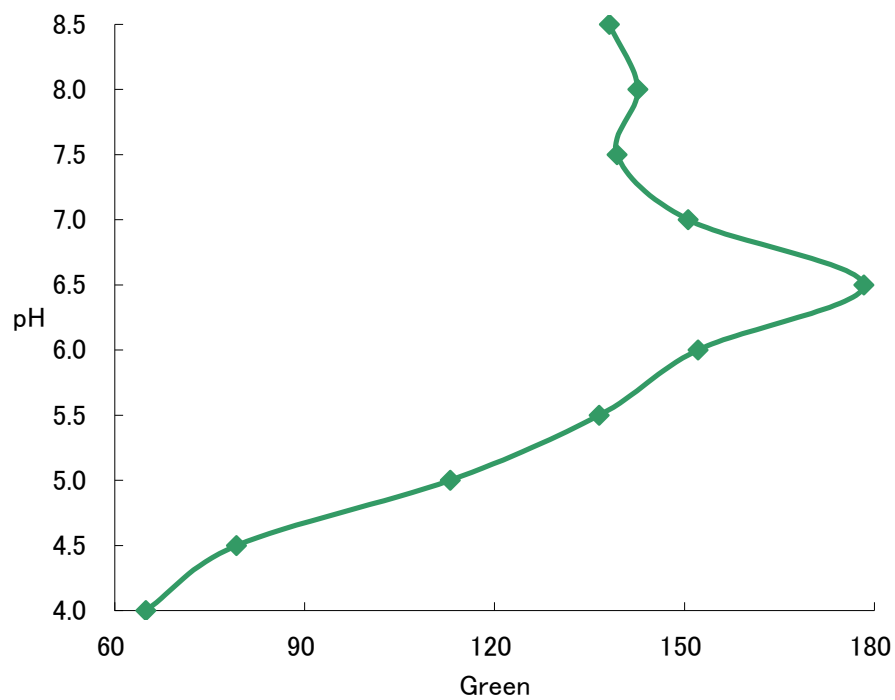


図 5.7 G-pH 線図 (土壌の 1 回目の実験結果)

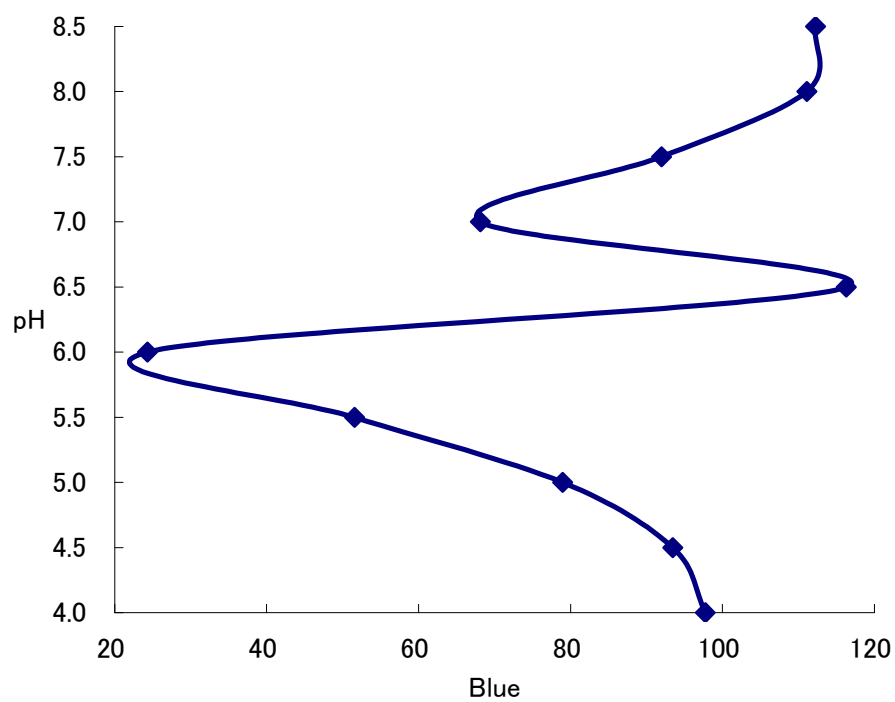


図 5.8 B-pH 線図 (土壌の 1 回目の実験結果)



11. これら R、G、B と pH の関係が単調関数となるように R・G の画像演算を行い、測定結果をグラフ化し、最小二乗法によって近似曲線を求める。

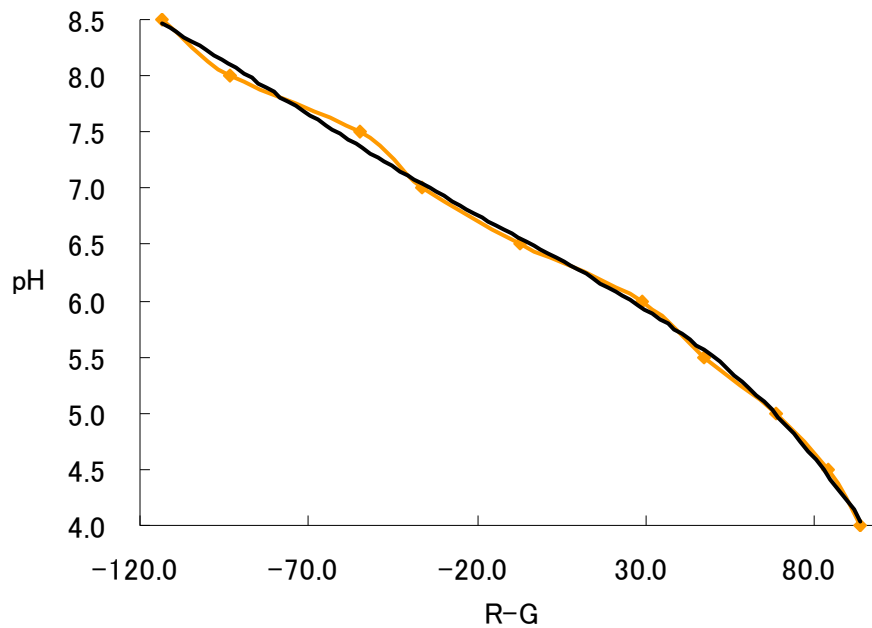


図 5.9 (R-G)-pH 線図 (土壌の 1 回目の実験結果)

$$y = -4 \cdot 10^{-9} x^4 - 6 \cdot 10^{-7} x^3 - 10^{-5} x^2 - 0.0163x + 6.435 \quad (\text{式 5.2})$$

y : pH

x : グレーレベル値

12. 近似曲線 (式 5.1) に試料の測定値を代入し、pH を推定する。

## 第4節 結果

表 5.2 から 5.4 に各試料の pH の測定結果を示す。

表 5.2 土壌の測定結果

	目測	ガラス電極法	ImageJ
1 回目	5.0	5.7	5.6
2 回目	5.0	5.8	5.3
3 回目	5.0	5.8	5.5
$x_{max}$	5.0	5.8	5.6
$x_{min}$	5.0	5.7	5.3
$R$	0.0	0.1	0.3
$\bar{x}$	5.0	5.8	5.5
$s^2$	0.0	0.0	0.0
$\sigma$	0.0	0.1	0.2

表 5.3 赤玉土の測定結果

	目測	ガラス電極法	ImageJ
1 回目	5.5	6.2	5.6
2 回目	5.0	6.3	5.5
3 回目	5.5	6.2	5.6
$x_{max}$	5.5	6.3	5.6
$x_{min}$	5.0	6.2	5.5
$R$	0.5	0.1	0.1
$\bar{x}$	5.3	6.2	5.6
$s^2$	0.1	0.0	0.0
$\sigma$	0.3	0.1	0.1

表 5.4 バーミキュライト

	目測	ガラス電極法	ImageJ
1 回目	5.5	6.2	5.7
2 回目	5.5	6	5.7
3 回目	5.5	6.2	5.9
$x_{max}$	5.5	6.2	5.9
$x_{min}$	5.5	6.0	5.7
$R$	0.0	0.2	0.2
$\bar{x}$	5.5	6.1	5.8
$s^2$	0.0	0.0	0.0
$\sigma$	0.0	0.1	0.1

$n$  : サンプル数

$x_{\max}$  : サンプルの最大値

$x_{\min}$  : サンプルの最小値

$R$  : 範囲  $R = x_{\max} - x_{\min}$  (式 5.2)

$\bar{x}$  : 平均  $\bar{x} = \frac{1}{n} \sum_{j=1}^n x_j$  (式 5.3)

$s^2$  : 標本分散  $s^2 = \frac{1}{n} \sum_{j=1}^n (x_j - \bar{x})^2$  (式 5.4)

$\sigma$  : 標準偏差  $\sigma = \sqrt{s^2}$  (式 5.5)

## 第5節 考察

まず、pH 値と RGB のそれぞれのグレースケール値の関係については、図 5.6 から図 5.8 に示すように 1 つのグレースケール値に対して複数の pH 値が得られる範囲があるため、適当な pH を求めることができない。とくに、図 5.8 に示すように Blue のグレースケール値ではその傾向が明らかにあらわれる。Red の場合 (図 5.6) は pH8.5 から 6.0 付近までは負の相関を示しているが、低い pH 域では濃度の pH 値依存性が低くなる。反対に Green の場合 (図 5.7) は pH4.0 から 6.0 付近までは正の相関を示すが、高い pH 域では依存性が低くなる。そこで、R-G の画像演算を行うことにより、図 5.9 に示すように両者の間に単調現象の関係が得られ、広い範囲の R-G の pH 依存性を確保することができた。

次に、各試料について、目測による方法の結果に着目すると、土壌とバーミキュライトでは測定値のばらつきが全くなく、赤玉土では 0.3 のばらつきがあることが分かる。これは、比色表の誤差範囲 R が 0.5 であることと判断の基準が測定者の視覚に依存していることによるものであり、きわめて不正確と言える。同様に、デジタル pH メータによる測定法は、いずれの試料でもばらつきの程度が 0.1 であることから精密な測定であると考えられる。また、ImageJ を用いた画像解析による測定は、測定値の範囲が 0.3 に収まっていることが確認でき、ばらつきの程度は土壌が 0.3、他の試料は 0.1 を示している。その原因は、土壌の 2 回目が 5.3 とやや低い測定値であることにあり、 $n=3$  というサンプル数の少なさからするとこれが確率的誤差であるかを判断することは難しい。

次に、ImageJ による測定法と目測による方法を比較すると、試験管内の溶液の色と比色表を対比することにおいては共通しているが、目測では誤差範囲 R が 0.5 であるのに対し、ImageJ による測定法では 0.1 で測定を行え、明らかな違いが伺える。すなわち、ImageJ による測定法は、目測の欠点を補うだけでなく、より高い精度で測定できることを意味する。同様に、ImageJ の測定値とデジタル pH メータの測定値を  $\bar{x}$  で比較すると、0.3 ないし 0.6 の誤差が生じている。この理由は、ImageJ による画像解析が簡易測定キットの比色表の撮影画像を用いて行っているためである。したがって、ImageJ による計測法とデジタル pH メータによる測定法 (ガラス電極法) を比較するには前提として予め、

- 1) 測定範囲の pH の検定試料液の調製を行い、ImageJ とデジタル pH メータで測定・比較する
- 2) 検定試料液をデジカメで撮影し、RGB 値のそれぞれについてグラフを作成し、R-G の画像演算で近似曲線を描く

という作業が必要であり、この作業を行うことによって系統誤差をなくすことができると判断した。

以上のことから、ImageJ を用いた画像解析による酸度測定法は、

- ・ 0.1 の誤差範囲で測定することができる
- ・ 人間の感覚量を定量化することができる

などの特徴を有し、今後、さらに多くの土壌について実験が蓄積されることによって酸度測定の 1 つの有効な手段になりうると考えられる。

## 第6章 結論

本研究では、教育現場における作物の生育への計測および制御技術を活用することを目的として、土壌の灌水制御および生育環境の計測を行った。

土壌の灌水制御は、土壌の乾燥状態に応じた自動灌水制御とネットワークを利用したリモート灌水制御の2通りの方法について試作・検討を試みた。土壌の乾燥状態に応じた制御では、灌水の必要性を判断する指標である  $pF$  値をセンサからの出力電圧に換算することで、**Arduino** を用いて電磁弁の開閉を制御することが示された。土壌の乾燥状態の把握には、採用する指標とセンサを設置する深さとの関係が生育環境に及ぼす相関についてしばしば課題として挙げられる。この課題を解決するために、土壌中の異なる深さにセンサを複数設置することが最も簡易な方法として考えられるが、圃場の広さや制御ボードのアナログ入力ポートの制限（**Arduino** のアナログ入力ピンは6本）などの条件から実用的ではない。水分状態の把握の解決方法として本研究では、灌水時間および待ち時間を設定し、一度の灌水で有効水分範囲に達しない場合に再度灌水を開始するようにプログラムで設計した。しかしながら、試作した灌水器を用いて実地での運用を行っていないため、今後、適切な灌水時間と待ち時間の設定を検討する必要があると考えられる。また、リモート灌水制御は、クライアントが灌水時間および灌水頻度を任意で決められる。ネットワークカメラなどと併用すれば、作物の生育管理の負担や苦手意識の軽減につながると考えられる。

生育環境の計測は、水分や気温などのリモート計測、画像計測による土壌酸度の測定を行った。**Processing** と **Arduino** による簡易計測システムを構築することによってネットワーク越しに計測項目をリアルタイムで表示させることができた。さらに、前述のリモート灌水制御と組み合わせることによって、**Processing** で土壌の水分状態を把握し、ネットワークを介して手動で灌水することや、新たにデータベースサーバを構築し、計測データを蓄積し、それらを任意で参照することで、より実用的なシステムとすることが可能となる。

画像計測による土壌の酸度測定では、**ImageJ** というオープンソースの画像処理ソフトウェアを用いて行うことで人間の感覚量に左右されずに濃度を定量化することができ、比較的高精度で計測できることが確認できた。今後、他の試料などでも実験を重ね、精度について検証していく必要がある。また、**ImageJ** を作物の生育環境の計測に活用した例として、デジカメで撮影したキュウリの葉から病害の範囲の割合を推定できることも確認している。

本研究を通して **Processing** と **Arduino** による制御環境、画像処理ソフト **ImageJ** が作物の生育環境の計測に有効に活用できることがわかった。いずれも優れた操作性を有していることから、教育現場で利用できる可能性があり、多くの分野や領域にわたって作物の生育管理や計測・制御にコンピュータの積極的な活用が期待できる。

## 参考文献および web サイト

### 第 1 章

- 1) 愛知県農業総合試験場 <http://www.pref.aichi.jp/nososi/>
- 2) 安東茂樹『中学校新学習指導要領の展開 技術家庭科技術分野編』、明治図書、2008 年 11 月
- 3) 文部省『中学校学習指導要領（平成 10 年 12 月）解説—技術・家庭編—』、東京書籍、平成 11 年 9 月
- 4) 建築発明工作ゼミ 2008 <http://kousaku-kousaku.blogspot.com/>

### 第 2 章

- 1) 犬伏和之、安西徹郎『土壌学概論』p79、朝倉書店、2001 年 4 月
- 2) 久馬一剛ほか『土壌の事典』、朝倉書店、1993 年 12 月
- 3) Arduino 公式サイト <http://www.arduino.cc/>
- 4) 株式会社タカギ公式サイト <http://www.takagi.co.jp/>

### 第 3 章

- 1) 建築発明工作ゼミ 2008 <http://kousaku-kousaku.blogspot.com/>
- 2) *Ben Fry* 『ビジュアルライジング・データ Processing による情報の視覚化手法』、オライリー・ジャパン、2008 年 11 月
- 3) Massimo Banzi 『Arduino をはじめよう』、オライリー・ジャパン、2009 年 3 月

### 第 5 章

- 1) 久馬一剛『最新土壌学』p89、朝倉書店、1997 年 3 月
- 2) シェファー/シャハトシャーベル『土壌学』p130、博友社、昭和 54 年 11 月
- 3) 菊池拓己、伊藤敏、井上祥史「画像解析による力学運動の計測」『日本産業技術教育学会第 51 回全国大会（仙台）講演要旨集』、p51、2008 年
- 4) 菊池拓己、伊藤敏、井上祥史「ImageJ による画像計測法の評価- デジタルカメラやビデオカメラを計測器として活用するために-」『岩手大学教育学部研究年報 2009』pp.65- 70、2009 年
- 5) ImageJ <http://rsb.info.nih.gov/ij/>

## 謝辞

本研究を進めるにあたり、多くの方々のご指導、ご助言、ご協力を賜りました。

指導教官の井上先生には、研究の進め方、論文の構成にいたるまで専門的なご指導ご助言を賜り、深く感謝致しております。

また、金澤先生や家政科の三井先生には土壌酸度の実験の進め方について、田中先生や宮川先生からは講義や学会発表で、前岩手大学教育学部教授の武田豊蔵先生や農場の佐藤さんには栽培管理全般について、それぞれご指導ご鞭撻を頂き誠に有難うございました。

そして他の所属の院生や学部生の方々など、研究にご協力頂いたすべての方々に深く感謝の意を表したいと思います。

## 付録資料

A1	作物の灌水開始時期	1
A2	作物の耐酸性	2
A3	自動灌水制御のプログラム	3
A4	リモート灌水制御のプログラム (クライアント)	4
A5	リモート灌水制御のサーバプログラム (サーバ)	6
A6	ネットワーク計測のプログラム (クライアント)	7
A7	ネットワーク計測のプログラム (サーバ)	10
A8	画像解析による力学運動の計測 (『日本産業技術教育学会第 51 回全国大会 (仙台) 講演要旨集』より)	11
A9	ImageJ による水平面内の運動解析 (『日本産業技術教育学会東北支部研究論文集 2008』より)	12
A10	ImageJ を用いた画像計測 (『岩手技術教育研究会誌No.38(2009)』より)	17
A11	Processing による簡易な制御教育環境 (『日本科学教育学会東北支部研究会研究論文集 2009』より)	23
A12	Arduino による栽培管理 (『日本産業技術教育学会第 27 回東北支部大会講演論文集』より)	25



# A1 作物の灌水開始時期

作物	かん水開始点の水分吸引圧 (深さ 10cm)	備 考
トマト	生育前期：pF2.0～2.4 生育後期：pF1.8～2.5	前期とは第3花房着果までで、多肥栽培では多かん水。 多水分管理では過繁茂、空どう果、奇形果、多腐れ果が多発しやすいので注意を要する。
メロン	定植活着期：pF2.0 前後 交配後：pF2.4 付近 ネット発生始めから完成期： pF2.4～2.7 以上 成熟期：pF2.5～pF2.7	日中は水分張力を低めに、夜間は高めに管理。収穫1週間前より水切りをして糖度を上げる。
キュウリ	生育前期：pF2.0～2.5 生育後期：pF1.7～2.3	前期少水分、後期多水分がよい。
ピーマン	pF1.5～2.0	
ナス	pF1.5～2.0	少かん水は、根系は多くはるが果実の光沢不良。
イチゴ	生育前期：pF1.5～2.0 生育後期：pF1.8～2.3	多肥栽培では低 pF(1.2)でかん水が必要。少水分管理では、小果となりやすいが糖度は高くなる。
セルリー	pF1.5～2.4	排水良好な所では土壌水分張力の変動幅を大きくしたほうがよい。

(位田 徳太郎、博友社「野菜の土壌生態・検定と肥培」47 頁より転載)

## A2 作物の耐酸性

---

1	ある程度の酸性が最適	イネ、サトイモ
2	最適 pH は弱酸性・中性だが、酸性に強い	サツマイモ、ダイズ、エンバク、ライムギ、ソバ
3	耐酸性が2に次ぐ	コムギ、ジャガイモ、ソラマメ
4	耐酸性が3に次ぐ	アズキ、トウモロコシ
5	酸性に弱い	オオムギ、エンドウ

---

(堀江 武 ほか、朝倉書店「作物学総論」126 頁より転載)

### A3 自動灌水制御のプログラム

```
int val;                                //使用変数の宣言

void setup(){
  pinMode(2,OUTPUT);                    //極性反転用リレー
  pinMode(3,OUTPUT);                    //電磁弁
  Serial.begin(9600);                  //シリアル通信開始 9600bps
}

void loop(){
  val=analogRead(0);                    //アナログ入力ピン 0 から val を読む
  if(val > 35){                          //pF2.3 以上の場合
    //灌水開始
    digitalWrite(2,LOW);                 //リレーOFF
    digitalWrite(3,HIGH);                //バルブ開
    delay(500);                          //電磁弁の通電時間 0.5 秒間
    digitalWrite(2,LOW);                 //リレーOFF
    digitalWrite(3,LOW);                 //バルブ閉
    delay(4500);                         //待ち時間
  }
  if(val < 19){                          //pF1.7 以下の場合
    //灌水停止
    digitalWrite(2,HIGH);                //リレーON
    digitalWrite(3,HIGH);                //バルブ開
    delay(500);                          //電磁弁の通電時間 0.5 秒間
    digitalWrite(2,LOW);                 //リレーOFF
    digitalWrite(3,LOW);                 //バルブ閉
    delay(4500);                         //待ち時間
  }
}
```

#### A4 リモート灌漑制御のプログラム (クライアント)

```
import processing.net.*;                                //ネットワークライブラリ

Client c;
String input;
int[] data={0,0};                                     //変数の宣言
int i,d0,d1;                                           //変数の宣言
PFont myFont;                                          //変数の宣言

void setup()
{
    size(320, 240);                                    //画面サイズの設定
    background(255);                                   //背景色を白にする
    fill(0,100,0);                                     //動作していないときの「OPEN」ボタンの色
    rect(20,20,40,20);                                 //「OPEN」ボタンの座標とサイズ
    fill(100,0,0);                                     //動作していないときの「OFF」ボタンの色
    rect(20,60,40,20);                                 //「OPEN」ボタンの座標とサイズ
    myFont=loadFont("Arial-BoldMT-35.vlw");            //使用するフォントおよびサイズ
    textFont(myFont,20);                               //テキストの表示位置
    c = new Client(this, "160.29.112.89", 80);          //サーバの IP アドレス
    c.write('o');
}

void draw(){
    fill(0);                                            //テキストの色
    text("OPEN",width/3,40);                          //テキストの表示位置
    fill(0);                                            //テキストの色
    text("CLOSE",width/3,80);                         //テキストの表示位置
}

void mousePressed(){                                  //マウスが押されたら通信開始
    //OPEN ボタンをクリックした場合
    if((mouseX >20) && (mouseX<60) && (mouseY>20) && (mouseY<40)){
        fill(0,255,0);                                //ボタンの色を明るい緑にする
        rect(20,20,40,20);
        c = new Client(this, "160.29.112.89", 80);     //サーバと通信開始
        delay(10);
        c.write('o');
        delay(300);
    }else{
        fill(0,100,0);                                //他の場所をクリックしたときの色
        rect(20,20,40,20);
    }
    //CLOSE ボタンをクリックした場合
```

```

if((mouseX >20) && (mouseX<60) && (mouseY>60) && (mouseY<80)){
    fill(255,0,0);                                //ボタンの色を明るい赤にする
    rect(20,60,40,20);
    c = new Client(this, "160.29.112.89", 80);    //サーバと通信開始
    delay(10);
    c.write('c');
    delay(300);
}else{
    fill(100,0,0);                                //他の場所をクリックしたときの色
    rect(20,60,40,20);
}
}

```

## A5 リモート灌漑制御のサーバプログラム (サーバ)

```
#include <Ethernet.h>                                     //イーサネットライブラリ
byte mac[] = { 0x0a, 0x0b, 0x0c, 0x08, 0xac, 0x01 };      //MAC アドレス
byte ip[] = { 160, 29, 112, 89 };                         //IP アドレス
Server server(80);                                         //ポート番号

void setup(){
  Ethernet.begin(mac, ip);
  pinMode(2,OUTPUT);                                       //リレー制御用ポート
  pinMode(3,OUTPUT);                                       //電磁弁制御用ポート
  server.begin();
}

void loop(){
  Client client = server.available();                      //サーバが受信可能なとき
  if (client) {
    while (client.connected()) {                          //クライアントと接続されている間
      if (client.available()) {                            //クライアントが送信可能ならば
        char c = client.read();                           //cを読み込む
        if (c == 'o') {                                    //バルブを開けるとき
          digitalWrite(3,HIGH);                            //バルブ開
          delay(500);                                       //0.5 秒間
          digitalWrite(3,LOW);                             //バルブ開の状態を維持
          break;
        }
        if (c == 'c') {                                    //バルブを閉めるとき
          digitalWrite(2,HIGH);                             //リレーON
          digitalWrite(3,HIGH);                             //バルブ閉
          delay(500);                                       //0.5 秒間
          digitalWrite(2,LOW);                             //リレーOFF
          digitalWrite(3,LOW);                             //バルブ閉
          break;
        }
      }
    }
    delay(1);
    client.stop();                                         //通信終了
  }
}
```

## A6 ネットワーク計測のプログラム (クライアント)

```
import processing.net.*;                //ネットワークライブラリを取り込む

Client c;                              //クライアントのインスタンス
String input;
int plotX1,plotY1;                     //変数の宣言
int plotX2,plotY2;                     //変数の宣言
float x,y;                             //変数の宣言
PFont myFont;                          //変数の宣言

int[] data={0,0};                      //変数の宣言
int i=1,d0,d1;                         //変数の宣言
float j,q,scl;

void setup() {
    size(720, 405);                    //画面サイズの設定
    //プロット領域の端の位置を設定
    plotX1=50;
    plotX2=width-plotX1;
    plotY1=50;
    plotY2=height-plotY1;
    smooth();
    background(224);                   //画面背景色の設定
    //プロット領域を白のボックスとして表示
    fill(255);
    rectMode(CORNERS);
    noStroke();
    rect(plotX1,plotY1,plotX2,plotY2);

    //フォントの指定
    myFont=loadFont("Arial-BoldMT-20.vlw");
    textFont(myFont,20);
    stroke(0);
    line(plotX1-2,plotY2,plotX1-2,plotY1);    // 縦軸

    //スケールの表示
    j=59;    //pF2.8
    scl=height-map(j,0,70,50,plotY2);        //値を増幅
    stroke(0);
    line(plotX1-10,scl,plotX1-2,scl);          //線の表示
    //値の表示
    fill(0);
    text("2.8",plotX1-40,scl-2);
}
```

```

j=50; //pF2.7
scl=height-map(j,0,70,50,plotY2); //値を増幅
stroke(0);
line(plotX1-10,scl,plotX1-2,scl); //線の表示

j=47; //pF2.5
scl=height-map(j,0,70,50,plotY2); //値を増幅
stroke(0);
line(plotX1-10,scl,plotX1-2,scl); //線の表示

j=34; //pF2.3
scl=height-map(j,0,70,50,plotY2); //値を増幅
stroke(0);
line(plotX1-10,scl,plotX1-2,scl); //線の表示

j=27; //pF2.1
scl=height-map(j,0,70,50,plotY2); //値を増幅
stroke(0);
line(plotX1-10,scl,plotX1-2,scl); //線の表示

j=23; //pF1.9
scl=height-map(j,0,70,50,plotY2); //値を増幅
stroke(0);
line(plotX1-10,scl,plotX1-2,scl); //線の表示

j=19; //pF1.7
scl=height-map(j,0,70,50,plotY2); //値を増幅
stroke(0);
line(plotX1-10,scl,plotX1-2,scl); //線の表示
//値の表示
fill(0);
text("1.7",plotX1-40,scl+10);

//サーバの IP アドレス、ポートの設定
c = new Client(this, "160.29.112.89", 80);
c.write("q");
}

```

```

void draw(){
  //軸ラベルの表示
  fill(0);
  text("pF",10,height/7);
  text("time(50ms)",(width/2)-70,height-10);

  if(millis()%500 == 0){

```



```

c = new Client(this, "160.29.112.89", 80);
delay(1);
c.write("q");

i=i+10;
i=i%(width-50);

}
if (c.available() > 0) {                                     //サーバからのデータ受信
    input = c.readString();
    print(input);
    if(input.indexOf("¥n")>5){
        data = int(split(input, ','));
        println(data[0]+" "+data[1]);
    }else{
        data[0]=d0;
        data[1]=d1;
    }
}

d0=data[0];
d1=data[1];

x=i+plotX1;                                                  //x 座標
y=height-map(data[0],0,70,50,plotY2);                      //y 座標

stroke(255,0,0);                                              //線の色
strokeWeight(3);                                              //線の太さ
line(x,plotY2,x,y);                                          //線の描画
stroke(255);
strokeWeight(3);
line(x+10,plotY2,x+10,plotY1);                               //消去

delay(50);                                                  //表示間隔
}

```

## A7 ネットワーク計測のプログラム (サーバ)

```
#include <Ethernet.h>                                //イーサネットライブラリ

byte mac[] = { 0x0a, 0x0b, 0x0c, 0x08, 0xac, 0x01 }; //MAC アドレス
byte ip[] = { 160, 29, 112, 89 };                    //IP アドレス
Server server(80);                                   //ポート番号

void setup(){
  Ethernet.begin(mac, ip);
  server.begin();
}

void loop(){
  Client client = server.available();
  if (client) {
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        if (c == 'q') {
          server.print(analogRead(0));                //アナログ入力値
          server.print(",");
          server.println(analogRead(1));               //アナログ入力値
          break;
        }
      }
    }
    delay(1);
    client.stop();
  }
}
```

(学生会員) 岩手大 ○菊池拓己

(正会員) 岐阜聖徳学園大 伊藤敏、岩手大 井上祥史

## はじめに

従来、落石など動体の空間座標の把握には複数のカメラを用いたシステムによる画像解析や、磁気あるいは光学センサによる計測が行われてきた。最近では市販のビデオカメラで運動体を撮影し、動画の解析から速度や加速度を求めることが教育現場で活発に行われるようになってきた。特にオープンソフトの ImageJ を用いた計測では、短時間で高い精度で運動体の空間座標を求められるためさまざまな試行評価ができるなど、教育利用に適した計測環境であることから、映像による力学計測法の特徴と可能性について評価することにした。

## 1. 運動体の撮影

教育利用を考え、弾むスーパーボールを斜め上から撮影して空間位置を求め、得られた重力加速度から計測誤差を求める単純な系で本計測法の特徴を評価した。簡単のためボールの軌跡が構成する  $x$ - $y$  面の  $x$  軸はカメラ座標系 ( $u, v$ ) の  $u$  軸と平行にした。そしてカメラと  $x$  軸との水平距離、水平面からの高さ、中央のカメラ視線が水平面と交わる点の位置、の 3 つの値を撮影条件として予め記録し、汎用の DV カメラでボールを撮影し avi ファイルとして PC に保存した。

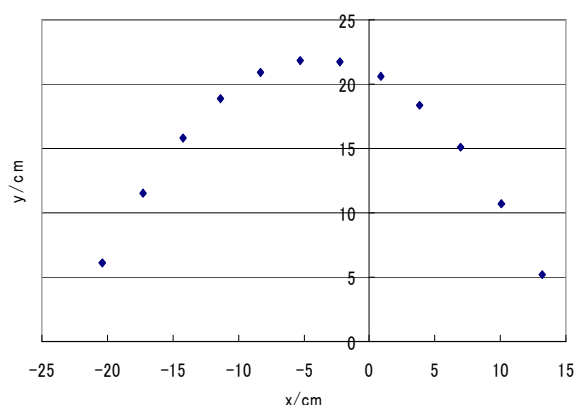


図1 画像計測で得られたボールの座標値

## 2. 画像解析

avi ファイルを非圧縮およびインターレース解除の後、ImageJ で RGB 分解して、さらにイメージの R-B 演算によりボールだけから成るスタックファイルを得た。さらに 2 値化処理を経て、重心位置、面積、楕円長半径、短半径を ImageJ で求め、その結果を csv ファイルとして Excel に取り込み、ボールの空間位置を求めた。図 1 に解析結果の一例を示す。

## 3. 画像計測の特徴

差分法で重力加速度を求めた結果、 $-975\text{cm/s}^2$  などの値が得られ、数回の実験でも 0.5% 以下の精度で重力加速度を測定できた。カメラ位置、視線中央位置など有効数字 2 桁であるため、本手法はほぼ正確な結果を与える画像計測法であるといえる。

束縛条件として  $x$  軸と  $u$  軸が平行ではなく、物体が水平面内に運動する場合には、さらに簡易に座標値を求めることができた。また束縛条件が無い一般の場合には、ボール画像の短半径の pixel 値を用いてカメラとボールまでの距離からボールの空間座標 ( $x, y, z$ ) を求めることができたが、その精度は前 2 者に比べて低かった。

この画像計測法は、(1) 解析できる運動体の速度の上限はカメラのフレーム速度に依存する、(2) 空間座標が求められる、(3) 解析精度が高い、(4) カメラだけの簡便な計測システム、(5) 面積または色情報によって複数の運動物体の識別が可能、(6) マクロ機能を利用して操作を簡略化できる、(7) カメラの焦点距離を計測したりスケールを映しこむようなキャリブレーションを必要としない、(8) カメラを運動体の中に設置して慣性力を計測することができる、などの特徴を持つ。教育利用をはじめ多くの分野での活用が可能である。

## ImageJ による水平面内の運動解析

## ImageJ analysis of motion in the horizontal plane

菊池 拓己\* 高城 大\* 井上 祥史\* 伊藤 敏\*\*

Kikuchi Takumi\*, Takagi Hiroshi\*, Inoue Shoshi\*, Itou Satoshi\*\*

水平面内の運動を ImageJ を用いて画像解析を行い、教材としての特性を評価した。斜め上方から汎用の DV カメラを用いて水平面内の運動を撮影したところ 1% 程度の誤差で位置解析が可能であった。この手法を用いて 1・2 次元のガラス球の衝突を解析して衝突時の運動メカニズムの推定を試みた。ImageJ を用いて汎用的な情報機器を実験の計測器として教育に活用する環境について考察した。

キーワード： ImageJ, 画像解析, 衝突, 運動量

## 1. はじめに

運動する物体の運動解析をするために、ストロボ写真やビデオカメラを用いて位置情報を求めることは古くから行われてきた。しかしながらこれらの手法は現像や手作業による位置解析などが必要で、即時性に乏しく手軽に実験を行うには困難な面があった。最近になって ImageJ に代表される画像解析のオープンソフトが利用できるようになり、ビデオカメラあるいは USB カメラとパソコンがあれば簡単な操作で動画画像を含む運動解析が行える環境が整ってきた。ところが今までの運動の解析法では、透視座標の修正を含まないようにできるだけ遠距離から視線軸に垂直な面内の運動を撮影しなければならないなどの制約が多かった<sup>1)</sup>。特に水平面内の運動を解析するには真上から撮影しなければならない、狭い実験室での運動解析には不向きな面があった。このため近距離でしかもカメラ視線軸と任意の角度をなす面内を運動する物体を画像解析できる方法が必要とされていた。そこで簡単な解析法を用いて水平面内のモデル運動の画像解析を行い、その精度を検証することにした。またその手法を用いて衝突メカニズムについて検討を試みた。

## 2. 画像による運動解析

## 2.1. ImageJ による運動解析

ImageJ は、NIH (米国立衛生研究所) が開発した画像処理のオープンソフトで、静止画像からピクセル演算により緑地面積の計算、粒子数カウントや粒子径の分散、クロマトグラフや溶液の濃度などを簡単に求めることができる。動画画像についても静止画のスタックとして扱い、一括して重心座標や軌跡などを求める豊富なツールが用意されている。

市販の DV カメラで撮影された動画画像は圧縮されたビットマップ・ファイルからなり、しかも走査線が間引かれたインターレース画像である場合が多い。このような avi 画像に対し、画素 (ピクセル) ごとの画像処理を行うためには、最初に非圧縮化およびインターレースモードの解除を行う必要がある。この処理を経て ImageJ に取り込んだスタックファイルは、一括して RGB 分解、ピクセル画像演算、2 値化処理を行うことができ、重心位置、面積、楕円長半径、短半径などを求め、その結果を csv ファイルとして出力できる。そして Excel に取り込まれた運動体の空間位置情報から、速度、加速度、エネルギーなどを求めること

(2009 年 7 月 18 日受理)

\* 岩手大学, \*\* 岐阜聖徳学園大学

† 2008 年 11 月 30 日仙台にて発表

になる。

## 2.2. 水平運動の撮影座標系

画像処理で運動解析を行う従来の手法は、透視座標系の修正を避けるために、遠くから運動体を撮影したり水平運動を真上から撮影するなどいくらかの制限が付いている場合が多い。そこで、この制限を取り外し、教室などのごく狭い範囲でも運動解析ができるように、斜め上からカメラを構えて垂直面内の運動を撮影し、ImageJで解析することにした<sup>2)</sup>。およそ90cm離れた斜め上方からボールの弾む垂直面内の運動を解析したところ重力加速度の平均値で $989\text{cm}\cdot\text{s}^{-2}$ が得られ、誤差が1%程度の計測法であることがわかった<sup>3)</sup>。この手法を拡張して水平面内の運動を斜め上方から撮影して運動解析を行うことにした。撮影系の幾何配置を図1に示す。

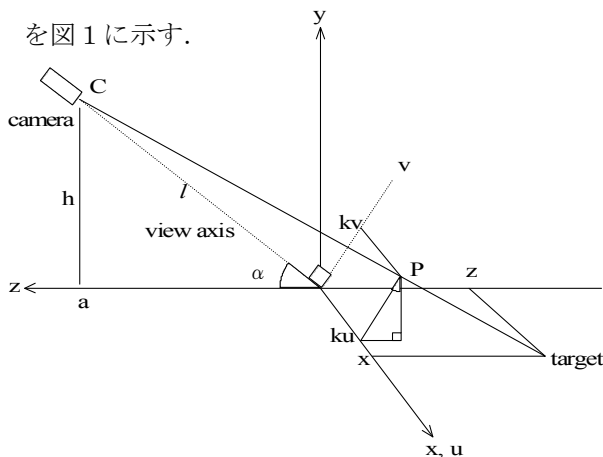


図1. 撮影系の幾何配置

図1のように水平面を $(x, z)$ 面、カメラの視線軸と直交する2次元画像座標系を $(u, v)$ 座標とし中心を原点とする。いま簡単のため $x$ 軸と $u$ 軸は重なっているとする。カメラ位置を $C(0, h, a)$ とし、カメラは座標原点と角度 $\alpha$ をなすものとする。

水平面内の座標 $(x, z)$ を求めるには、実験の準備としてカメラ位置から $\alpha = \tan^{-1}(h/a)$ 、 $x$ 軸上に置いたスケールの画像のピクセル値から画像座標と実座標との比例定数 $k$  (cm/pixel)を決定しておく。画面の水平方向と垂直方向で縦横比が異なる場合には、 $u, v$ 軸方向の比例定数 $k_u, k_v$ を予め求めておく。

運動体のボールをターゲットとすると、ター

ゲットとカメラを結ぶ視線と $(u, v)$ 面との交点は $P(ku, kv \cos \alpha, -kv \sin \alpha)$ となるので、直線CPの方程式は

$$\frac{x-0}{ku-0} = \frac{y-h}{kv \cos \alpha - h} = \frac{z-a}{-kv \sin \alpha - a} \quad (1)$$

水平面内の運動を扱うので $y=0$ から

$$x = \frac{-kuh}{kv \cos \alpha - h} \quad (2)$$

$$z = \frac{kv \sin \alpha + a}{kv \cos \alpha - h} h + a \quad (3)$$

を得る。

## 3. 計測結果と考察

### 3.1. 水平面内の円運動

水平面内の運動解析例として、円運動の軌跡の位置精度を評価した。赤いボールに糸をつけ、一端をテーブルに固定し、糸のたるみを取った状態でボールを水平面内で接線方向に投げて半径30cmの円運動をする様子を斜め上から撮影し、画像解析で重心位置を求めた。DVカメラで撮影したaviファイルの非圧縮化にはVirtualDub.exe、インターレースモードの解除にはAviUtl.exeを用いた。スタックファイルをRGB分解の後、R-Gの画像演算を経て2値化して、重心座標 $(u, v)$ を求めExcelで空間位置 $(x, 0, z)$ を得た。実験条件は $a = 98\text{cm}$ ,  $h = 50\text{cm}$ ,  $k = 0.128\text{cm/pixel}$ であった。水平面内のボールの1/30secごとの重心位置の解析結果を図2に示す。

図中の実線は半径30cmの円弧で、円の中心から重心位置までの距離と紐の半径30cmとのずれの割合は平均で-0.09%、誤差の標準偏差は0.86%であった。カメラ位置など実験条件の測定値の有効数字は2桁であることを考えると、水平面内の運動解析は精度よく行われていることを表している。

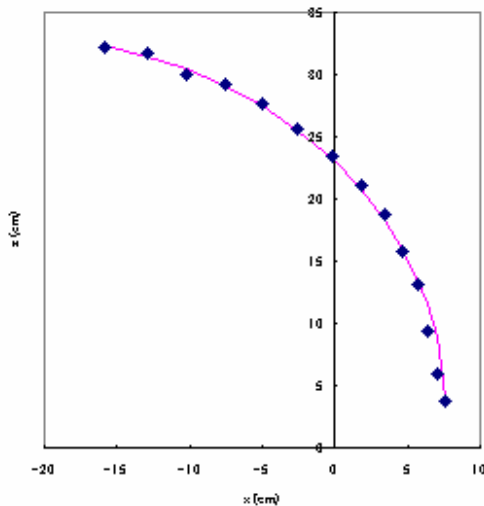


図 2. 水平面内の円運動の軌跡.  
実線は半径 30cm の円弧.

### 3.2. レール上の 2 球の衝突

レール上で 2 つのガラス玉を衝突させ、衝突過程を調べるために衝突前後の速度変化を調べた.

最初に同じ大きさ (質量  $m=6.2\text{g}$ , 半径  $r=0.84\text{cm}$ ) の小球同士を衝突させた場合、図 3 に示すように衝突直後に両球とも速度の絶対値はピークを経て徐々に減少する. 表 1 に衝突前後のエネルギー変化を示す. 速度が変化する理由は、球がレール上を転がりながら移動して衝突すると衝突直後には移動する向きと回転の向きが逆転するため、衝突前の運動エネルギー  $E_k$  から衝突前の回転エネルギー  $E_r$  のおよそ 2 倍近くが失われ、大きく速度が減少するものと考えられる. このとき衝突後の運動方向と逆向きの回転が摩擦によって速度方向に順応するまでに時間がかかるため速度にピークが現れるものと考えた.

衝突前の運動エネルギーが衝突前と後の回転エネルギーと衝突後の運動エネルギーに分配されると仮定すれば、ガラス球の衝突係数を  $e$  として

$$E_k' + E_r' + E_r = E_k e^2 \quad (4)$$

が成り立つ.  $E_r = 2/5 \cdot E_k$ ,  $E_r' = 2/5 \cdot E_k'$

の関係から表 1 の値を使うと  $e = 0.76$  を得る.

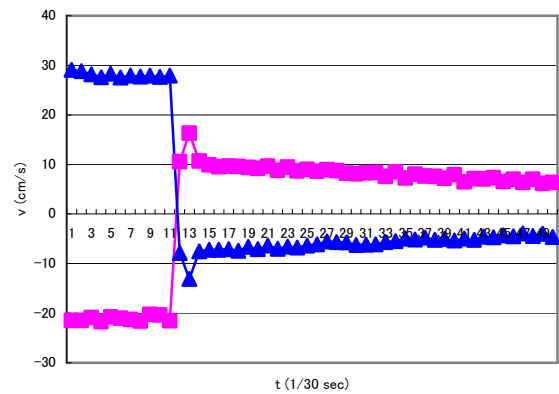


図 3. レール上の小球同士の衝突の速度変化

表 1 小球同士の衝突前後のエネルギー変化.

値は 2 球の合計エネルギーの 5/30sec の平均値.

(単位は  $10^{-6}\text{J}$ )

	回転 energy	運動 energy	全 energy
衝突前	$E_r=1506$	$E_k=3765$	5272
衝突後	$E_r'=186$	$E_k'=465$	651

### 3.3. レール上の大小球の衝突

次に小球( $m=6.2\text{g}$ ,  $r=0.84\text{cm}$ )と大球 ( $M=21.1\text{g}$ ,  $R=1.25\text{cm}$ ) のガラス玉をレールの上で衝突させた場合の速度変化を図 4 に、衝突前後のエネルギー変化を表 2 に示した.

この衝突では、小球大球ともに衝突直後に速度が増加している. これは小球の周速度が大球より大きいいため衝突によって小球の周速度が反転すると共に大球の周速度も増加するためと考えられる. 従って両球とも進行方向の速度より回転の周速度が速く、摩擦で順応するに従って速度が増加するものと考えられる. 衝突後の大球の運動エネルギーの最小値が 0 と仮定し、衝突前の全運動エネルギーが衝突前の大球の回転エネルギーと衝突後の小球の全エネルギーに分配されたとすると

$$(E_k + E_R) e^2 = E_R + E_k' + E_r' \quad (5)$$

これより表 2 の値を代入して  $e = 0.76$  を得る.

この値は小球同士の衝突で得られたガラスの反発係数と同じである。

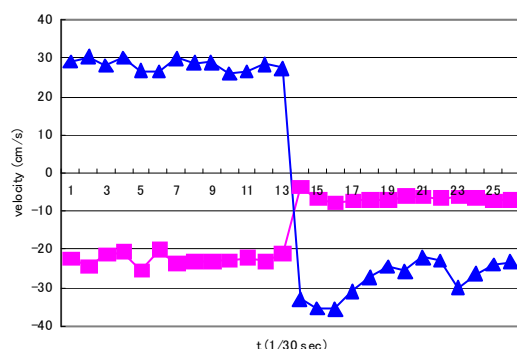


図 4. レール上の大小球の衝突過程.

▲ : 小球, ■ : 大球.

表 2 小球と大球の衝突前後のエネルギー変化.  
値は衝突前後それぞれ 4/30sec の平均値. 単位は  $10^{-6}\text{J}$ .

	回転 energy	運動 energy	全 energy
衝突前	$E_r=930$	$E_k=2310$	小球 3260
	$E_R=2135$	$E_K=5340$	大球 7470
衝突後	$E_r'=840$	$E_k'=2100$	小球 2630
	$E_R'=206$	$E_K'=513$	大球 720

また, 小球の速度は衝突後に長い周期の振動が見られる. これは半径の異なる大小球の衝突軸が上下方向に傾くために, 衝突後に上下方向に振動が与えられすべり摩擦力に変動が現れるためと考えている.

### 3.4. 水平面内の大小球の衝突

次に, 水平面内の運動解析として, ガラス板の上で大小 2 つのガラス玉の衝突を扱った. ガラス球はガラス板の上をレールと同じように滑ることなく転がる. 図 5 の小球の速度変化から  $v_x$  におい

てレール上の小球同士の速度変化と類似の現象が起っているものと推測できる. 衝突前の全運動エネルギーが衝突後の全エネルギーと衝突前の小球の回転エネルギーに分配されるとすると, 表 3 の記号を使って

$$(E_k + E_K)e^2 = E_r' + E_R' + E_k' + E_K' + E_r \quad (6)$$

と表され,  $e = 0.75$  となる. また, 衝突における運動量変化の大きさは小球で  $370\text{g}\cdot\text{cm/s}$ , 大球で  $340\text{g}\cdot\text{cm/s}$  となった. 両者の差は回転エネルギーの寄与の程度によるものと考えられる.

このように衝突過程には衝突条件に依存した複雑なプロセスが絡んでいる可能性があり, 詳細な検討

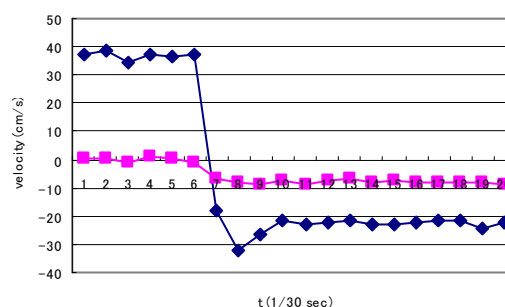


図 5. 大小球の衝突による小球の速度変化

◆ :  $V_x$ , ■ :  $V_z$

表 3 衝突前後のエネルギー変化 (単位は  $10^{-6}\text{J}$ )  
値は衝突前後 5/30s の平均値.

		回転 energy	運動 energy	全 energy
衝突前	小球	$E_r = 1670$	$E_k = 4170$	6840
	大球	$E_R = 3390$	$E_K = 8480$	11870
衝突後	小球	$E_r' = 860$	$E_k' = 2140$	3000
	大球	$E_R' = 690$	$E_K' = 1730$	2420

を行うためには高速な撮影による解析が不可欠である。最近の市販デジタルカメラの中には高速で撮影できるものがあり容易に入手可能である。このように ImageJ を用いると身の回りで起る複雑な過程からなる現象も画像解析の対象となり、教育材料として利用範囲が広がる。

#### 4. 画像計測の教育利用

汎用の情報機器とオープンソフトの ImageJ を用いる画像計測環境は、教育現場に導入しやすく、また近距離の撮影でも水平面内あるいは垂直面内の位置解析が精度よく行える特徴を持っている。

また ImageJ による画像解析は処理内容をリアルタイムで画像として確認できるため、学習者は画像解析の内容を直感的に把握することが可能で教育利用に適している。高校生を対象としたアンケートでも、一連の画像解析の操作で困難と感じるものは見られず、むしろ理解しやすかったとする回答が多かった。

また ImageJ による画像計測法は、(1)解析できる現象はカメラのフレーム速度に依存する、(2)斜め上方から近距離の撮影で水平面内の運動を解析できる、(3)面積や色情報によって複数の運動体を識別することができる、などの特徴を持つ。このように ImageJ を用いる画像計測法は様々な教育需要に柔軟に対応できるため、多くの教育場面での活用を期待したい。

#### 5. 参考文献

- 1) 浅岡 末治, 永谷 恵一, 平田 充: 家庭用ビデオカメラを利用した運動の記録  
<http://www.asahi-net.or.jp/~uu9m-hrt/DVvideo/video.htm>
- 2) 菊池 拓己, 伊藤 敏, 井上 祥史: ImageJ による画像計測法の評価-デジタルカメラやビデオカメラを計測器として活用するために-, 岩手大学教育学部研究年報, 第

68 巻, pp65-69,(2009)

- 3) 菊池 拓己, 伊藤 敏, 井上 祥史: 画像解析による力学運動の解析, 日本産業技術教育学会第 51 回全国大会講演要旨集, p51,(2008)



## 1. はじめに

市販のデジタルカメラやDVカメラなどの画像記録機器は、汎用性を満たすために厳密な規格化が図られている。扱いやすい汎用の画像解析ソフトを使用することによってこれらの汎用機器を身近な精密測定器として実験や実習に活用することができる。このような要求を満たすソフトウェアとしてImageJを用いた場合の解析精度や操作性などの特徴を、簡単な力学運動系をサンプルとした運動解析を行うことによって評価した。そしてImageJを用いることによってデジタルカメラやビデオカメラが中高等学校の科学実験の汎用の計測機器として、また、リモートセンシング画像を用いてImageJで自然環境解析が行える可能性があるなど、多分野にわたる活用法を示した。

## 2. ImageJについて

ImageJはNIH（国立衛生研究所）のWayne Rasbandらによって開発された無償で利用できる画像解析ソフトであり、各種の画像解析機能と分かりやすいユーザインターフェースを備えている。前身はSionImageで、葉や細胞の面積、そしてDNAなどのクロマトグラフから成分濃度を求めるなど主に生物分野の研究用としてApple Machintoshで使用されてきた。ImageJは画像の加工を行いながらも、基本的にはカラー画像のRGB分解、2値化を行って様々な処理をする画像処理ソフトである。運動解析を行うには、動画像をaviファイルとして取り込み、すべてのフレームのスタック画像を一括してRGB分解、2値化、画像の差分演算を行い、解析した演算結果を時系列のCSVのテキストファイルとしてExcelにそのまま渡すことができる。このような機能を持つImageJは、画像処理をする目的とプロセスが明確なため解析目的に合わせて様々な解析手段を検討することができる柔軟な画像解析ツールとして有効に機能する。

## 3. 垂直面内の運動解析

動体の空間座標の把握には複数のカメラを用いたシステムによる画像解析や磁気或いは光学センサによる計測が行われてきた。教育現場においては、ストロボや扱いやすさの点から市販のビデオカメラを使用して運動体を撮影し、動画像を解析して速度や加速度を求めるなどの取り組みが多く報告されている。これらの運動体の解析は、比較的遠距離からの撮影や装置の点から制限が多い。

そこで、ImageJの画像解析機能のチェックとして最初に試みたのが垂直面内における力学運動の解析である。弾むスーパーボールを斜め上から撮影して空間位置を求め、得られた重力加速度から計測誤差を求める単純な系で本計測法の特徴を評価した。簡単のためボールの軌跡が構成する $x$ - $y$ 面の $x$ 軸はカメラ座標系 $(u,v)$ の $u$ 軸と平行にした。そしてカメラと $x$ 軸との水平距離、水平面からの高さ、中央のカメラ視線が水平面と交わる点の位置、の3つの値を撮影条件として予め記録し、汎用のDVカメラでボールを撮影しaviファイルとしてPCに保存した。図1に座標系を示す。

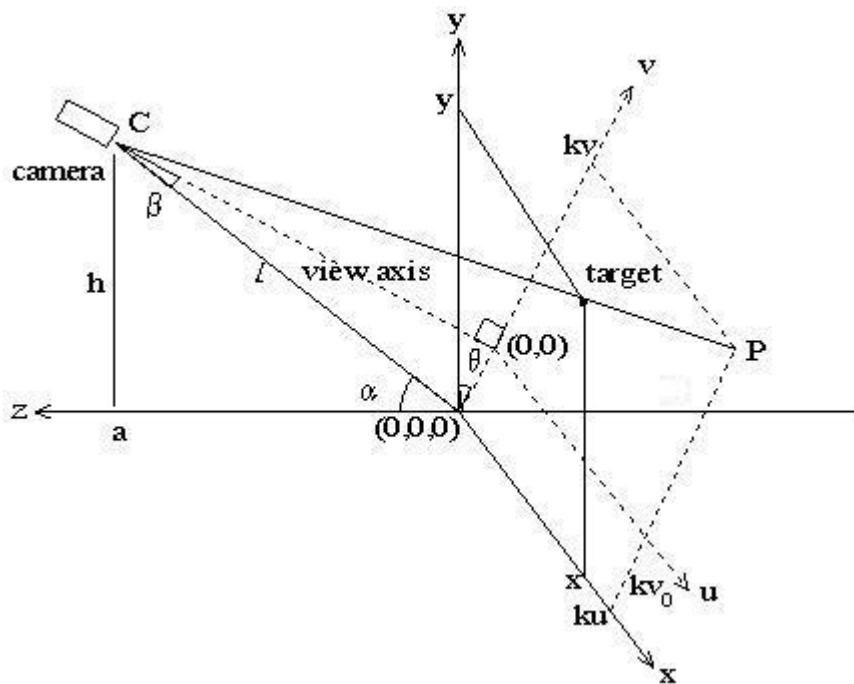


図 1 垂直面内の画像座標系

avi ファイルを非圧縮およびインターレース解除の後、ImageJ で RGB 分解し、さらにイメージの R・B 演算によりボールだけから成るスタックファイルを得た。さらに 2 値化処理を経て、重心位置、面積、楕円長半径、短半径を ImageJ で求め、その結果を csv ファイルとして Excel に取り込み、ボールの空間位置を求めた。図 2 に解析結果の一例を示す。

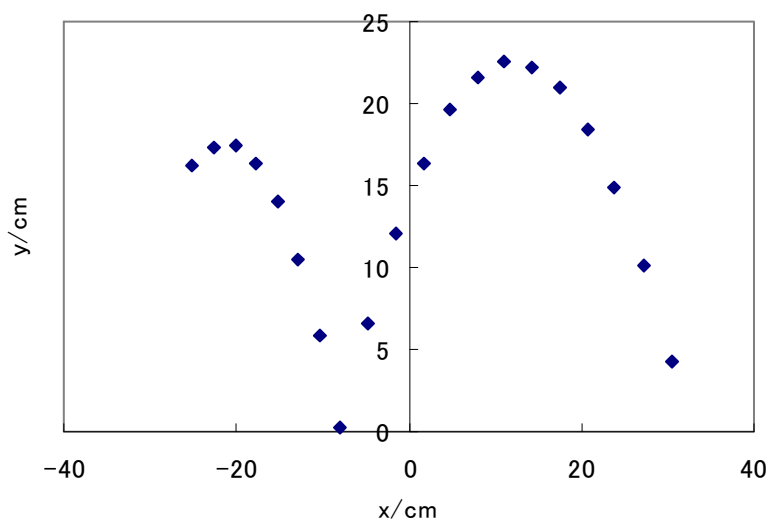


図 2 垂直面内の力学運動の解析結果

差分法で重力加速度を求めた結果、 $-975\text{cm/s}^2$ などの値が得られ、数回の実験でも1%以下の精度で重力加速度を測定できた。カメラ位置、視線中央位置などは有効数字2桁であるため、本手法はほぼ正確な結果を与える画像計測法といえる。運動解析の実験として、このほかに水平面内の運動やビー玉の衝突によるエネルギーの損失などの実験を行った。いずれも比較的高精度で速度、加速度などの物理量を計測することができた。このように汎用のDVカメラとImageJによる運動解析は、教育に有効利用できること

を確認できた。

#### 4. 衛星画像による自然環境解析

力学運動の解析のほかに、衛星画像による自然環境解析に本手法を用いた場合について考察した。人工衛星や航空機などに搭載されたセンサによって、さまざまな現象を広い範囲にわたって調査するリモートセンシングは 1970 年代から行われてきたが、最近では簡易に衛星画像を得ることが出来るようになってきたため、衛星写真から森林面積を求めることを試みた。サンプルとして図 3 に示すように盛岡市の衛星写真を Google Earth で取得して用いた。なお、距離と pixel 数の関係は 15.1km で 189pixels であった。



図 3 盛岡市の衛星写真（Google Earth より）

不必要な部分をカットし、盛岡市の概形を取り出し、画像の 2 値化を行った（図 4）。植生を調べるのに使われる正規化植生指数（NDVI : Normalized Difference Vegetation Index）は、以下の式で表わされる。

$$NDVI = \frac{NIR - RED}{NIR + RED}$$

NDVI は、植物の葉が赤色光（RED）を吸収し、近赤外電磁波（NIR）を強く反射する性質を利用した指数である。NIR データが入手できないため、緑の植生を調べるのは RED の吸収を利用した近似を用いることにした。盛岡市の衛星写真を RED の値のみで 2 値化すると、図 4 において、黒い部分が緑地に相当する。

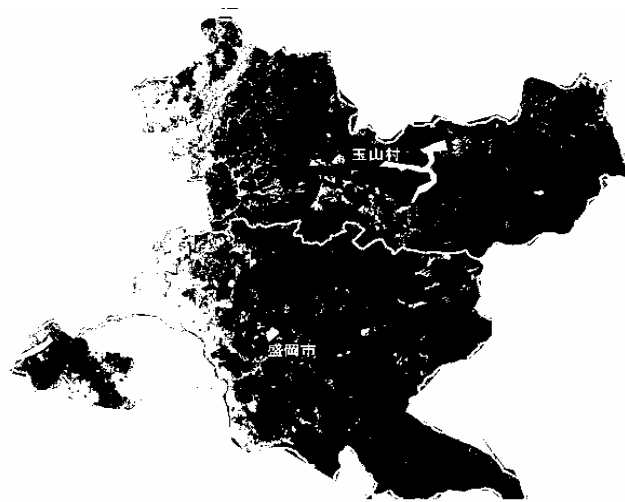


図4 2値化した盛岡市の衛星写真（緑の部分が黒に相当）

2値化した画像の黒い部分の面積を求めた結果、緑地面積が 110,989pixels となり、 $\text{km}^2$ に換算すると  $708.45\text{km}^2$  となった。盛岡市によると、玉山村と合併した平成 18 年度の統計で市全体の森林面積は  $65091\text{ha}$  即ち  $650.91\text{km}^2$  であることから、 $57.54\text{km}^2$ 、割合にしておよそ 10%の計測誤差が生じている。誤差の原因としては、

- （１）市の概形を大まかに決定したこと
- （２）北上川をはじめとする市内の河川の面積を含んでいること
- （３）統計年度と衛星写真の日時のずれ
- （４）耕作緑地のカウント

などが考えられる。これらの誤差要因を考慮したうえで衛星写真を利用した緑地面積などの環境計測も手軽に行えることがわかる。

## 5. 画像解析のその他の応用例の提案

### 5.1 赤外線写真

図5に赤外線カメラで撮影された風景写真を示す。



図5 赤外線カメラで撮影された風景写真とカラー写真  
(出展「福原のページ(植物形態学・生物画像集など)」から)

図 5 から、

- ・ 青空や海・川などは赤外線を吸収するので黒く写る。
- ・ 雲は赤外線をより拡散するため白く写る。
- ・ 針葉樹より広葉樹の方が反射率が高く白く写る。

ということが分かる。したがって、赤外線航空写真を入手できれば、ImageJ で画像解析を行うことで広葉樹か針葉樹かの識別をすることができる。また例えば、植物の樹冠高を求めるには樹木の葉の高さによって、リグニン含量に直接的变化があることを利用して、衛星写真から森林の高さを推定することが行われている。同様の解析手法を用いることで植生分布の把握や植物群落の分布地図あるいは稲作の病虫害などの把握に役立つ可能性がある。

参考までに以下に航空写真や衛星写真の主な入手先を示す。

- ・ 国土情報 Web マッピングシステム <http://nlftp.mlit.go.jp/WebGIS/index.html>

1974 年(昭和 49 年度)から 1990 年(平成 2 年度)にかけて国土地理院が撮影したもので国内全域における約 40 万枚のカラー写真がある。無償。

- ・ (財) 日本地図センター <http://www.jmc.or.jp/>

空中写真の大まかな撮影範囲と撮影年度を調べることができる。有償。

- ・ (社) 日本森林技術協会 <http://www.jafta.or.jp/index-j.html>

森林の管理のために山間部を中心に撮影されたものが多くある。有償。

- ・ ランドサット衛星画像 <http://www.kashmir3d.com/landsat/>

可視画像や赤外線画像など、膨大な画像データが蓄積されている。無償。

このほかにも民間企業によって撮影・提供されたものもある。

## 5.2 パックテスト

パックテストは、水質検査や環境調査に利用される比色分析法の 1 つで、溶液の色調またはその色の濃さから化合物の濃度を定量する分析法である。水の汚れを示す代表的な指標として BOD (Biochemical Oxygen Demand) と COD (Chemical Oxygen Demand) がある。

BOD は、主に河川の汚濁指標として用いられ、COD は湖沼や海域の指標として用いられる。

パックテストはこれらの BOD と COD を求める手軽な目視法であるが、測定者の視覚によって誤差が発生しやすい。パックテストの色をサンプルと比色するとき ImageJ を活用することができる。

## 5.3 スペクトル解析

次に、光のスペクトル解析を行った。太陽光を白紙で反射し、それを光源として撮影した画像の分光画像を得た (図 6)。



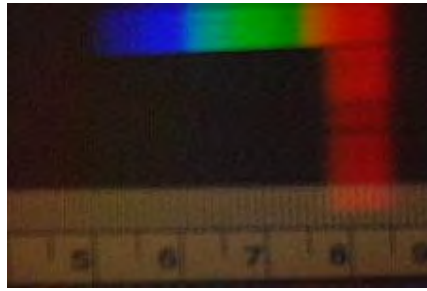


図6 分光画像

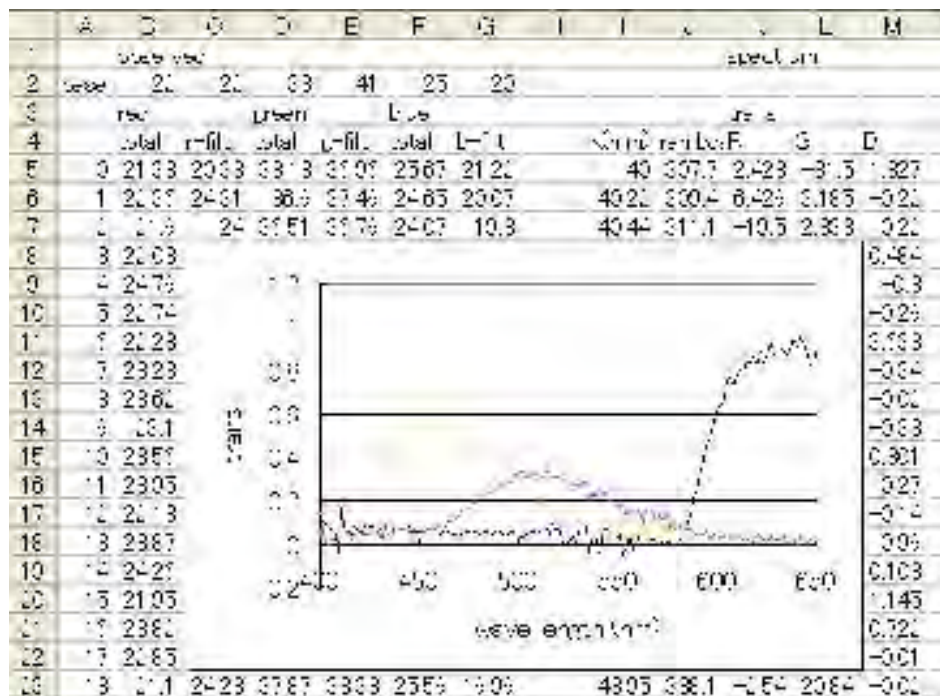


図7 赤・緑・青色の亚克力板の透過率の波長依存性

図7は、赤青緑の3色のカラーの亚克力板の透過率のスペクトルを描いたものである。このように簡易な回折格子があれば、分光器として使用できることが確認することができた。

## 6. おわりに

ImageJによる運動体の画像計測法は、

- (1) 解析できる運動体の速度の上限がカメラのフレーム速度に依存する
- (2) 空間座標が求められる
- (3) 解析精度が高い
- (4) 汎用のカメラだけの簡易計測システム
- (5) 面積や色情報によって複数の運動体を識別することができる
- (6) カメラの複雑なキャリブレーションをする必要がない
- (7) 解析に使用するソフトウェアがすべてオープンソースであること
- (8) 計測結果が直接 csv ファイルとして得られること

などの特徴を持つ。力学運動の解析や面積計算のほかに、ImageJで溶液の濃度解析や光のスペクトル解析なども行えることを確認している。このようにImageJは中学校や高等学校における多分野にわたる画像計測法であり、今後の教育利用が期待される。

## Processing による簡易な制御教育環境

### Simple Environment of Control Education using “Processing”

菊池 拓己

伊藤 敏

井上 祥史

KIKUCHI, Takumi

ITOU, Satoshi

INOUE, Shoshi

岩手大・教（院）

岐阜聖徳学園大

岩手大

Iwate University Gifu Shotoku Gakuen University Iwate University

**【要約】** プログラム開発環境の Processing と Arduino 制御ボードを用いて簡易に制御教育が行えることを示した。プログラミング経験のない高校 1 年生に対してネットワーク制御まで含む実習を行ったところ、7 割以上の生徒が興味を示し、生活環境の中の制御を意識し、視野を広げることができた。

**【キーワード】** Processing, Arduino, プログラミング, 制御

#### 1. はじめに

従来のプログラミングや制御の学習の多くは、プログラミング言語やコマンドの習得に時間を割かれ、アルゴリズムの考察を育む段階にまで深められないため敬遠されることが多かった。しかしながら、学習指導要領の改訂により中学校技術科ではプログラミングや制御が必修化され、情報教育の中で行う効率的で効果的なプログラミングや制御の学習方法の開発が求められている。そこで、画像処理など高度な処理まで簡易に開発できる Processing の環境に注目し、Arduino 制御ボードを用いて制御教育を行い、その効果を評価することにした。

#### 2. 制御環境

##### (1) Processing

Processing は、MIT で開発されたオープンソースで、数値計算から 3 次元の描画まで行うことのできるプログラム開発環境である。シリアル通信機能や画像のピクセル処理も可能で簡易に画像解析も行うことができる。Arduino<sup>1)</sup> というハードウェアボードと組み合わせてアナログ・デジタルデータを処理することができ、また汎用の USB-I/O ボードのコントロールも可能である。作成したプログラムはアプレットとしてブラウザで閲覧できるなどの教育用のプログラム開発環境として優れた特徴を備えている。

##### (2) Arduino

Arduino は 6 チャンネルの AD 変換ポートと 14 ビットのデジタル IO を持つワンボードマイコンと USB シリアル通信チップから成る。Arduino 開発環境を利用して USB ケーブルを接続したままコンパイルしたコードを書き込むことができる。さらに、Ethernet ボードを使用することにより、Arduino をサーバとして機能させ、ネットワーク制御を行うことも可能である。

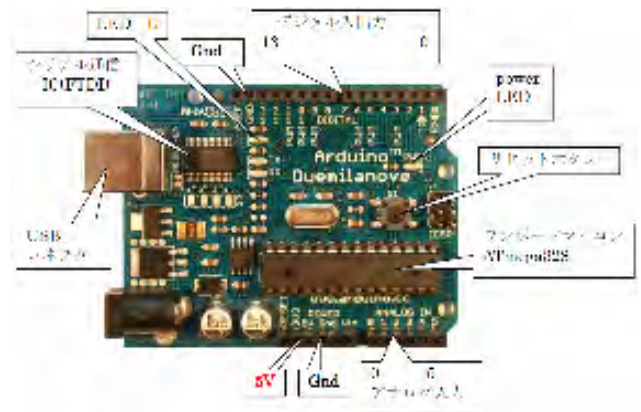


図1 Arduino ボード

#### 3. 制御実習

プログラミング経験のない高校 1 年生 20 人を対象に、以下の項目について 6 時間の制御実習を行った。

##### 1) 8 ビット LED の点灯制御

Processing の操作に慣れるため最初の 30 分

間にマウス操作を含む簡単な実習を行った。その後、図2に示す系でマウスによる LED の点灯制御プログラムを作成した。

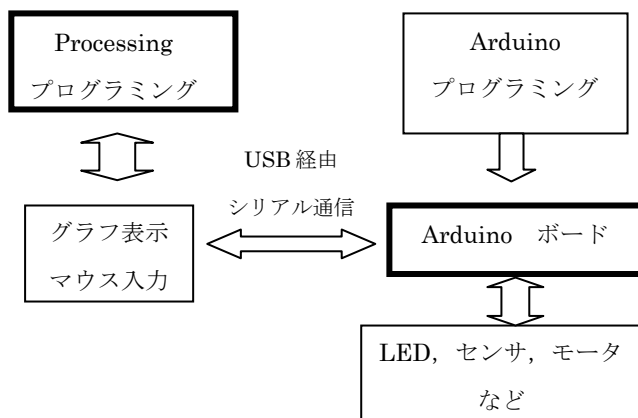


図2 制御系の概要

## 2) 3次元加速度センサを用いた制御<sup>2)</sup>

次に、加速度センサを用いて3次元空間の中で仮想物体を動かした。Arduinoのアナログ入力ピンに接続した3次元加速度センサから得られるx,y,z方向の加速度をProcessingにシリアル転送して仮想物体の位置を変化させるもので、センサを傾けて仮想空間を移動することから生徒の興味をひきつけることができた。

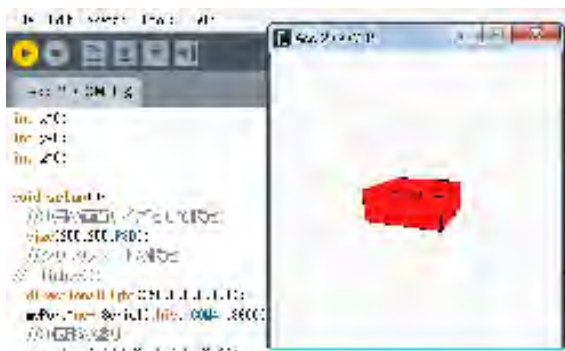


図3 加速度センサによる  
仮想物体のコントロール

## 3) ネットワーク制御

Processingのネットワーク機能を利用して、5関節アームロボットのネットワーク制御を行った。生徒の代表者をクライアントにして、ロボットのアームや指を遠隔操作した。これにより携帯電話でもネットワーク制御が簡単に行えることを生徒は実感していた。

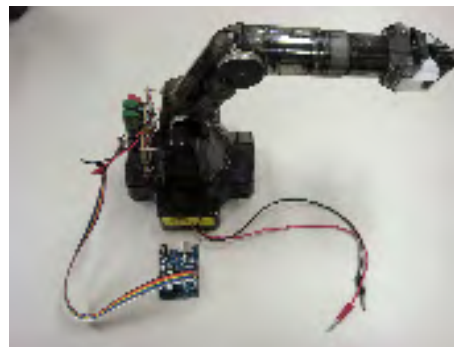


図4 アームロボットとArduinoボード

## 4. 制御環境の評価

ProcessingとArduinoを用いた実習後に記述式のアンケートを実施した。その結果、Processingに関しては全体の80%の生徒が「楽しい・おもしろい」と回答し、Arduinoに関して印象に残った事柄を聞いたところ、「LEDの点滅の仕方を変えるのがおもしろかった」が30%、「センサで3D画像を制御したのがおもしろかった」が25%であった。その他の感想として「信号や電光掲示板が動くしくみが分かった」「ゲームの作り方が分かった」といった実社会や実生活と結び付けた回答があり、全体としておよそ7割以上の生徒が興味・関心を示した。この結果から、ProcessingとArduinoによる制御環境が、プログラミング初学者にとって受入れられやすく、アルゴリズムの考察に集中できることが分かった。

今回実習したProcessingプログラミング・制御の教育環境はオープンソースであり、USBを通して電源が供給されるため、教育現場での管理も容易で導入しやすい。そして、教育効果も期待できるため、今後は中学生を対象とした授業実践やAppletの利用、環境計測への利用などで実践を重ね、新しい制御カリキュラムの提案を展望していく。

## 参考文献

- 1) Massimo Banzi, Arduinoをはじめよう, オライリー・ジャパン, 2009年
- 2) 建築発明工作ゼミ

<http://kousaku-kousaku.blogspot.com/>



菊池拓己（院）\*

伊藤 敏\*\*

井上祥史\*

作物の栽培に伴う環境計測や土壌の水分管理を省力化することを目的に、簡易に導入できる IO ボードの Arduino とそのプログラミング環境である Processing を用いて遠隔で計測データを取得および灌水できるシステムの開発を行った。そして、教育利用に向けてのシステムの評価を行った。

キーワード：Arduino、Processing、栽培、制御

## 1. はじめに

作物の生育の計測項目として、気温、湿度、日射量、土壌水分、土壌温度などが挙げられる。従来、これらの計測項目を頻繁に観察・記録することは困難であったが、フィールドサーバの導入により、生長の定点観測や病害中の発生などの圃場のモニタリングや環境計測などが容易に行えるようになった。また、土壌の水分管理には、タイマー制御や水分センサを利用して自動的に灌水することも行われている。教育現場でこれらのフィールドサーバやリモート管理を簡易に行うことができるシステムとして Arduino を用いた環境計測や自動灌水システムを試作し、評価することにした。

## 2. Arduino による土壌水分量の制御

Arduino を用いて作物に最適な土壌水分量を維持する散水制御装置の開発を試みた。また、ネットワークを通してリモートで圃場の環境計測および制御を行った。

### 2.1 Arduino サーバ

Arduino は ATMEGA328 というワンボードマイコンが組み込まれた入出力ボードで、14 本のデジタル IO ピンと 6 本のアナログ入力ピンがある。オープンソースで提供されている Arduino IDE でプログラムを作成し、USB 経由で Arduino ボードに書き込んでサーバとして独立に動作する。一方、リモートのクライアントには簡易に導入できる Processing を用いて時系列グラフの表示および電磁弁のリモートコントロールを行った。図 1 に作製したシステムの概要を示す。このシステムは Arduino サーバによる自動灌水と Processing による環境計測とリモート灌水を主な目的としている。

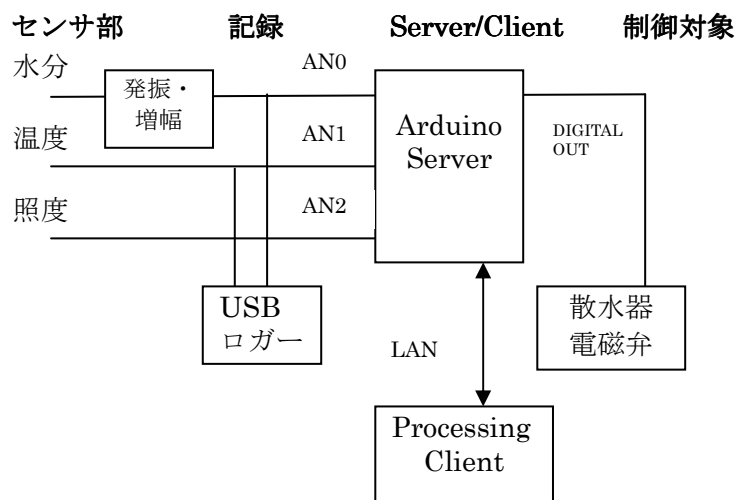


図 1 システムの概要

### 2.2 制御回路

試作した制御基板の回路図を図 2 に示す。水分量の測定は、ステンレス製の 2 本の電極間の交流インピーダンス（200Hz）の変化から求めた。図 4 に土壌水分量と製作した回路の出力電圧との関係を示す。

Arduino のアナログ入力機能は 10 ビット幅の分解能で作成した水分センサから得られた入力電圧によって電磁弁を制御する。散水器は市販のタカギ G216 を使用し、その電磁弁は、消費電力を抑えるため、信号が来て弁が動作した後、その信号が遮断されてもその状態を保持する構造になっている。このため、リレーを用いて矩形波を 2 つのポートに与えて極性を変化させることにした。図 3 に概要を示す。水分量が減少すると散水を開始し、一定時間(T<sub>1</sub>)散水した後、十分に水分が土壌中にしみ渡った時間(T<sub>2</sub>)後に再び測定を開始することとした。T<sub>1</sub>、T<sub>2</sub> は散水口とセンサ位置、圃場の広さなどによって異なる。また、矩形波の幅は 100ms とした。

\* 岩手大学教育学部

\*\* 岐阜聖徳学園大学経済情報学部

† 2009 年 11 月 29 日、東北支部大会

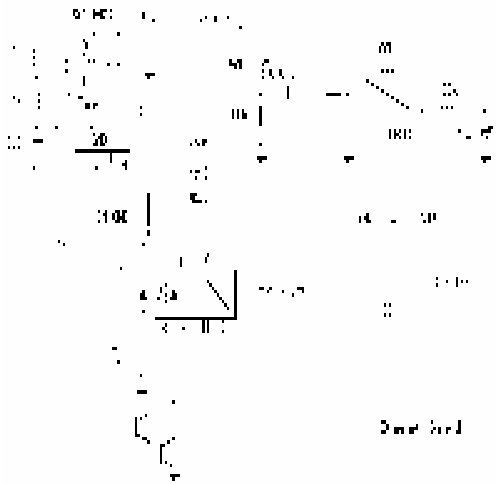


図2 制御回路

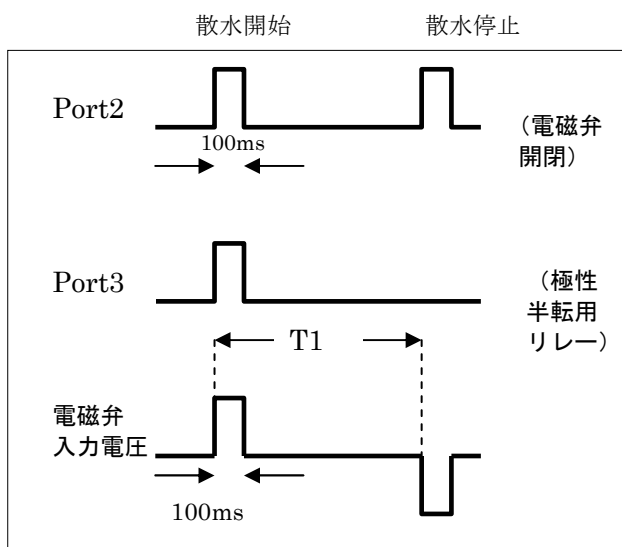


図3 電磁弁のタイミングチャート

### 2.3 水分量の計測

まず、市販の pF メータ（大起理化学工業株式会社製 DIK-8343）を用いて土壌の水分状態を計測した。pF 値は、作物が土壌中の水を吸い上げるのに必要な力を示し、灌水の必要性を判断する指標であり、土壌が乾燥状態のときは値が高くなり、水分過多のときは低くなる。pF 値と作成した回路の出力電圧の関係を図4に示す。pF 値が 2.2 以上になると散水を開始し、pF 値が 2.0 以下になると散水を停止した。

## 3. 計測データの取得

### 3.1 データロガー

USB データロガー（LASCAR 製の EL-USB-2）

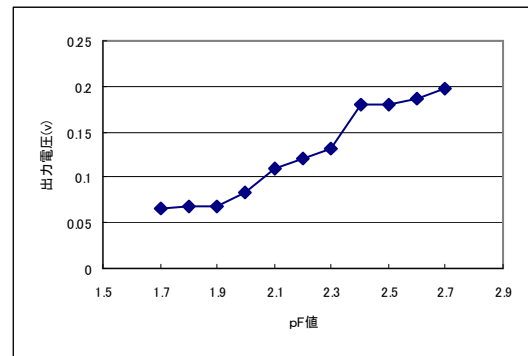


図4 センサ回路出力電圧の pF 値依存性

で水分量および温度のデータを記録するとともに水分量、照度、地温の計測値をブラウザおよび Processing でグラフ表示させた。

### 3.2 ネットワークによる計測

図1に示すように、Arduino を計測サーバーとし、Processing をクライアントとして水分量および気温の時系列変化を表示させた。Arduino には Ethernet Shield と呼ばれる Ethernet ボードを付加して用いた。また、Processing 上でボタンをクリックすると手動でリモート散水を可能とする機能を付加した。

## 4. おわりに

Arduino をサーバとして自動灌水およびリモート計測・制御システムが簡易に作成可能であることが示された。教育現場で導入することは容易であると考えられる。

一方、Processing を常時起動してデータを記録していくことは現実的ではないため、データベースサーバを別に設けて Web ブラウザからのリクエストに応じて任意の期間の水分量などのデータをグラフ表示できることが望ましい。このため現在データベースサーバの構築と表示プログラムの作成を行っている。また今回の報告で、計測データは水分量と温度に限定したが、必要に応じて照度、地温、湿度、そして画像なども測定対象になるため、より機能を充実したフィールドサーバの構築を検討している。

## 参考文献

- 1) Massimo Banzai・船田 巧: Arduino をはじめよう, オライリー・ジャパン
- 2) 建築発明工作ゼミ  
<http://kousaku-kousaku.blogspot.com/>