平成 20 年度卒業論文

論文題目

センサネットによる侵入者経路検知

神奈川大学 工学部 電気電子情報工学科
学籍番号　200502738
山田　高明

指導担当者　木下宏揚　教授

目 次

図 目 次

表 目 次

1 序論

今日の社会発展に伴い、セキュリティシステムも様々な形式が考案され社会に浸透している。金銭はもちろん重要な情報に対するセキュリティの重要性は年々増している。最近では凶悪犯罪の増加に伴い企業だけではなく一般の家庭においてのセキュリティの重要性も再確認する必要性がある。セキュリティとは危険な状態から守り、安全を保障する事となっているが大きく防犯と防災に分けられ、防犯は守る対象により以下の様に使い分けれらる場合がある。

- 警護 ···· 人
- 警備 ···· 場所、建物、企業、貴金属や人も含む
- 保安 ···· 航空、鉄道
- 治安 ···· 社会
- 安全保障 ···· 国家

この中の警備に着目する。一般的に事務所などの警備業務対象施設に警備員が常駐する常駐警備が行われてきたが夜勤業務となることが多いため人件費の増大などから現在では監視カメラなどを設置したセキュリティシステムなどの導入も増え、機械を使った警備が主流となっている。

しかし、防犯カメラによるセキュリティシステムにも人によっては圧迫感を与えてしまう事やカメラの台数や性能に大きく依存している事等の問題も考えられる。そこで低コストで導入できる光の環境の変化を感知し検知対象の検知及び進入経路検知を可能にするシステムを考案する。

光の環境の変化の感知には光によって光起電力を発生させる半導体素子であり、フォトダイオードよりも光起電力の大きい超高輝度 LED をセンサとして使用する。センサに検知対象が接近したときにできる影により、光の環境の変化が光起電力の電圧降下として表れることを利用する。

センサは μ IP[10] を実装した AVR チップ [2][3][4] と組み合わせ、イーサネットモジュール [8] を通し TCP/IP[5][6][7][11] で通信可能にすることでホストコンピュータとネットワークで通信し感知したデータを送信させるようにする。さらにセンサを複数用意し任意の場所に設置することでセンサネットを構築し時間差で表れる電圧降下から侵入者の経路検知を可能にすることを目的とする。

2 基礎知識

2.1 μ IP

μ IP は Swedish Institute of Computer Science, the Networked Embedded Systems group, Adam Dunkels 氏 [9] によって組み込み向けに開発された TCP/IP スタックであり 8bit のマイクロコントローラ [3][4] で動作するように実装されているフリーソフトである。合わせて 15 の設定ファイルやアプリケーションファイル [1] から構成されており、ユーザは設定ファイルに設定されている IP アドレス [5] を変更したり、アプリケーションファイルを変更するなど自分の開発環境に合わせて加工することが可能になっている。全部合わせてもきわめてサイズが小さくソースコードはコメントが行き届いており、組み込み用途のみならず TCP/IP スタックの実装を学ぶ上でも有効なものである。

特徴 [10]

- ソースコードは文章化とコメントが付けられよく説明されている。
- とても小規模のコードサイズである。
- RAM の使用率が低く、コンパイル時に変更可能である。
- ARP,SLIP,IP,UDP,ICMP(ping),TCP/IP プロトコルである。
- Web server,Web client,e-mail sender(SMTP slient),Telnet sever,DNS hostname resolver などアプリケーション例が含まれている。
- 同時に複数の TCP 接続が可能であり、接続の最大数はコンパイル時に変更可能である。
- 複数接続されていても受動的に受信できる、最大数はコンパイル時に変更可能である。
- 商業、非商業において自由に利用できる。
- RFC が発行されており、TCP/IP プロトコル、fragment reassembly、retransmission time-out estimation が実装済である。

2.2 プロトコル

人間は知能、応用力、理解力を持っているため、特に意識しなくても意思の疎通を図ることができるがコンピュータによる通信の場合ではそうはいかず、物理的なレベルからソフトウェアのレベルまで多くの部分で明確な規約を決めなければならない、通信中に起こる可能性のある問題に対しても適切な処理を行わせる必要がある。このようなコンピュータ同士の通信においてコンピュータ同士できめ細かく決められている規約が「プロトコル」[5][6][7][11] である。

2.2.1 TCP/IP の歴史

TCP/IP は現在、もっともよく使用されているプロトコルであり、TCP/IP をサポートしていない OS はほとんど販売されていない。これほどまでに TCP/IP がサポートされるようになった主な経緯には以下の事が関係している。

- パケット交換技術、パケット通信
- ARPANET の誕生
- UNIX の普及

1960 年代後半、アメリカの組織 DoD(the Delartment of Defense) を中心に通信技術の開発が行われた。そこでは複数のユーザで同時に 1 つの回線を共有し、回線の利用効率を向上させるパケット交換技術、パケット通信が注目されパケット通信の試験を行うため大学と研究機関の 4 つのノードを結んだネットワークが開発された。そこに一般ユーザーを取り込んだ、当時としては非常に大きなネットワークである ARPANET が誕生した。ARPANET 内の研究グループによりただのパケット交換だけでなく信頼性の高い通信手段として開発されたのが TCP/IP である。また、当時はすでに大学や研究機関で内部に TCP/IP が実装されている BSD UNIX が広く利用されていたため 1983 年には ARPANET の正式な接続手順として TCP/IP が採用され、同年には Sun Microsystem 社が TCP/IP を実装した製品を一般ユーザー向けに提供し始めた。インターネットは、終端ノード間の UNIX のコンピュータ同士をつなげる形で普及してきたため、コンピュータネットワークの主流と言える TCP/IP は UNIX と密接な関係を持って発達し普及、世の中にサポートされてきた事になる。[5][6]

2.2.2 TCP/IP プロトコル階層モデル

TCP/IP は TCP と IP という 2 つのプロトコルだけではなく、IP や ICMP、TCP や UDP、TELNET や FTP、HTTP など TCP や IP に深く関係するプロトコルが多く含まれており、インターネットを構築するうえでの必要なプロトコルがセットになっている。

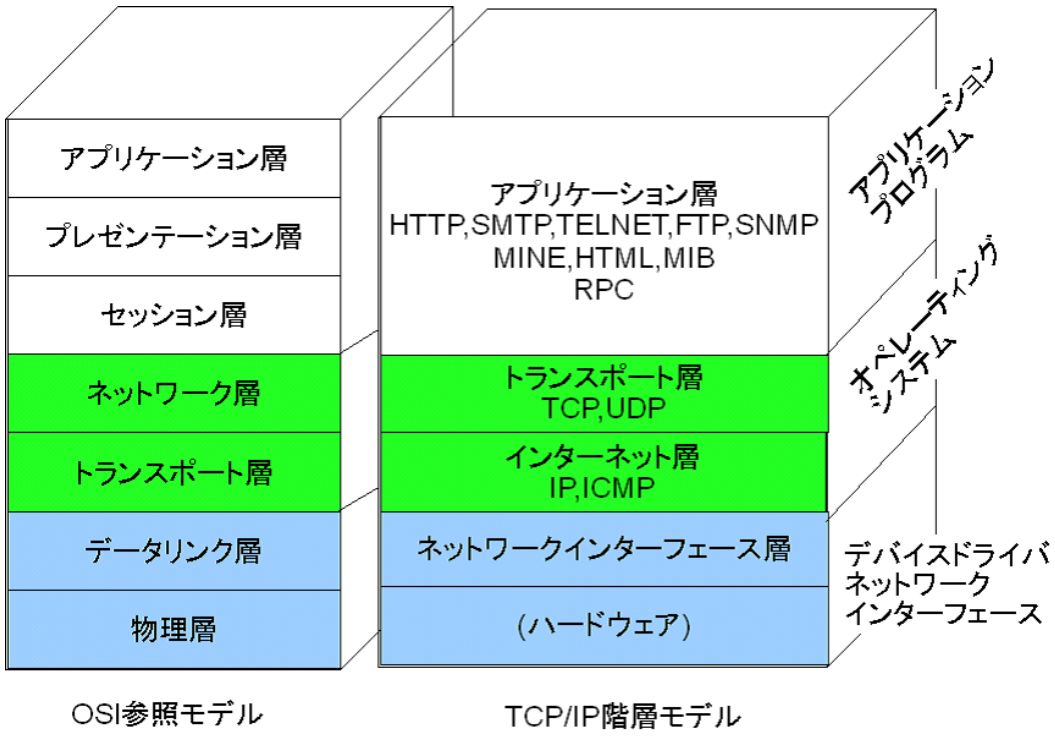

図 2.1: 階層モデル

2.2.3 ネットワークインターフェース層

ネットワークインターフェース層とは一般に「デバイスドライバ」と呼ばれるものである。デバイスドライバは OS とハードウェアの橋渡しをするソフトウェアこれらを利用するコンピュータの OS にインストールすることでネットワークインターフェースを利用できる環境が整う。広く共通化が進んだハードウェアでは、OS 内部に標準ドライバが含まれている場合が多い。標準ドライバがサポートしないハードウェアに関しては、一般に、そのハードウェアを提供するメーカがデバイスドライバを製品に添付するか、あるいはインターネット上で配布する。

2.2.4 インターネット層

インターネットそうでは、IP プロトコルが使われる。これは OSI 参照モデル [5][6] の第三層であるネットワーク層の役割と同様である。TCP/IP 階層モデルではインターネット層とトランスポート層は OS の内部に組み込まれている事を想定している。また、ルーターはインターネット層を利用してパケット転送する機能を実装しなければならない。IP(Internet Protocol) はネットワークをまたいでパケットを配送するためのプロトコルであり、これによりパケットを世界中に届けることが可能になっている。IP はデータリンクの特性を隠す役割もあり、通信したいホストの間の経路がどのようなデータリンクになっていようとも、通信を可能にする。IP は信頼性のないパケット交換だがインターネット全体にパケットを送り届けるためのプロトコルであるまた、ICMP(Inter Control Message Protocol) は IP パケットの配布中に何らかの異常が発生してパケットを転送できなくなった場合に、パケットの送信元に異常を知らせるために使われるプロトコルでありネットワークの診断などにも利用される。

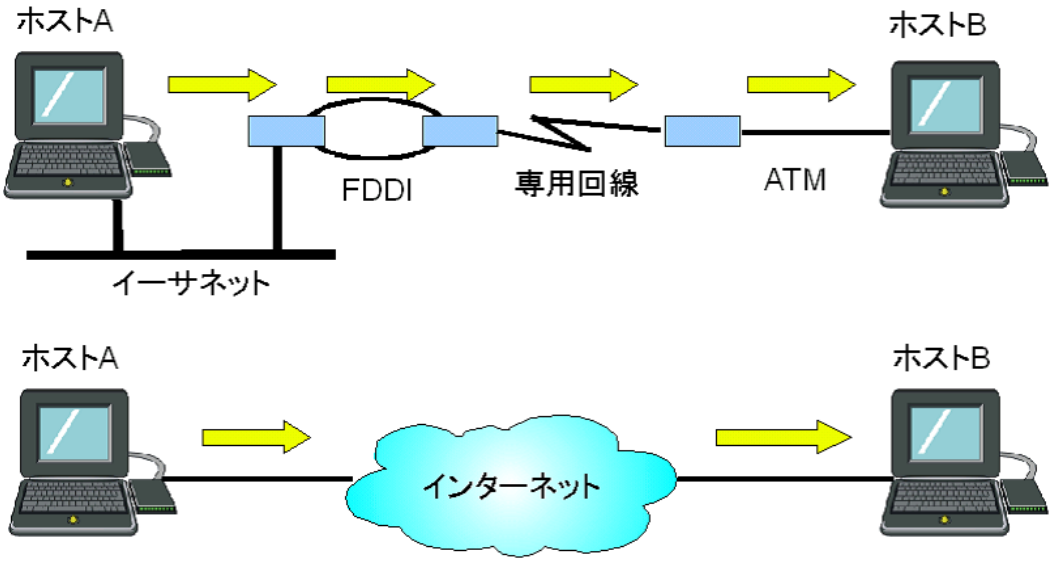

図 2.2: IP のパケット配送

インターネット層によってネットワークの細かい構造が抽象化される。両端のホストからは通信相手のコンピュータが雲のようにモヤモヤとしたネットワークの延長上に接続されているイメージとなる。インターネットはインターネット層の機能を備えたネットワーク。

2.2.5 トランスポート層

TCP/IP には 2 つのトランスポートプロトコルがあり、基本的には OSI 参照モデルのトランスポート層の役割を持っている。またアプリケーションのプログラム間の通信を実現するのが TCP/IP のトランスポート層の役割でもある。コンピュータでは複数のプログラムを同時に動作させることができるが、どのプログラムとどのプログラムが通信しているか、プログラムを識別するためにポート番号と言うアドレスを利用し識別するのがトランスポート層の役割である。TCP/IP 階層モデルではこのトランスポート層も OS の内部に組み込まれている事が想定されているがそれは通信を行っているアプリケーションプログラムを識別し、そのプログラムとデータの受け渡しをするためのサービスを提供するのは OS の役割と考えられているからである。TCP/IP の 2 つのトランスポート層のプロトコルを下記する。

- TCP(Transmission Control Protocol)
  TCP はコネクション型であり終端間でデータの到達を保障する信頼性のあるトランスポート層のプロトコルである。経路の途中でデータがなくなったり順番が入れ替わったとしてもうまく解決することができる。また、ネットワークの帯域幅を有効に利用する工夫や混雑を和らげる工夫がなされている。しかし、コネクションの確立と切断だけで 6〜8 つものパケットが飛び交うためデータの総量が少ない場合には無駄が多くなり、ネットワークの利用効率を向上させるための仕組みが組み込まれているため音声や映像データなどのように一定間隔で決められた量のデータを送信するのには向いていない。
- UDP(User Datagram Protocol)
  UDP は TCP とは異なり信頼性のないコネクションレス型のトランスポート層のプロトコルです。相手に届かない場合や相手のコンピュータがネットワークに接続されていないといったケースの場合、UDP は相手からの送信したデータが届いたかどうかの応答の確認は行わず、確認はアプリケーションのプログラムが必要に応じて行う。UDP はデータの量が少ない場合や、ブロードキャストやマルチキャストの通信、ビデオや音声などのマルチメディア通信に向いている。

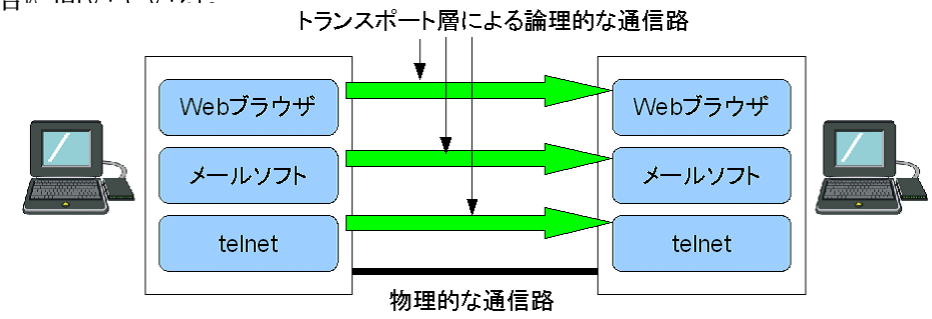

図 2.3: プログラム間での通信

2.2.6 アプリケーション層

TCP/IP 階層モデルでは OSI 参照モデルのセッション層、プレゼンテーション層、アプリケーション層は全てアプリケーションプログラムの中で実現されていると想定されている。単一のアプリケーションの内部にそれぞれの機能が実装される場合や複数のアプリケーションプログラムに分けて実装されることもある。TCP/IP のアプリケーションプログラムの機能には OSI 参照モデルのアプリケーション層の機能だけでなくセッション層の機能やプレゼンテーション層の機能が含まれている。殆んどの TCP/IP のアプリケーションはクライアント・サーバモデル [5][6] で構成されている。サービスを提供するプログラムがサーバで、サービスを受けるプログラムがクライアントである。この通信モデルでは、サービスを提供するサーバプログラムはいつクライアントから要求が来ても対応できる様にあらかじめホスト上で動作させておかなければならない。

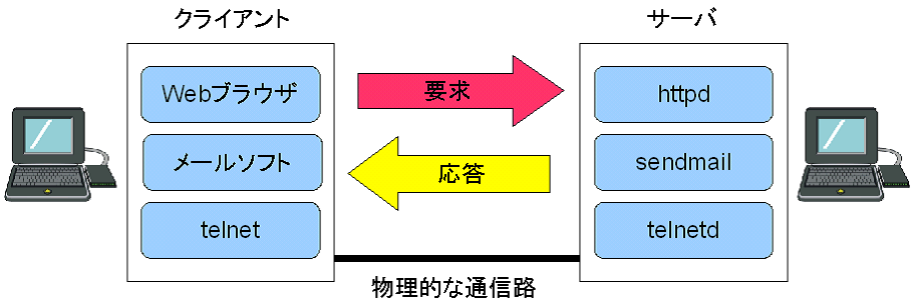

図 2.4: クライアント・サーバモデル

代表的な例として WWW(World Wide Web)[5][6][7] を挙げるとユーザはマウスやキーボードで「ブラウザ」というソフトを操作しブラウザを通してネットワーク上のサイトなどを閲覧、情報発信ができる。ブラウザとサーバ間の通信で使われるプロトコルは HTTP(HyperText Transfer Protcol) で送信に使われるデータフォーマットは HTML(HyperText Markup Language) である。この場合、HTTP が OSI 参照モデルのアプリケーション内で通信に関係しているアプリケーション層に該当し HTML が転送するデータの表示管理を行うプレゼンテーション層のプロトコルに該当することになる。

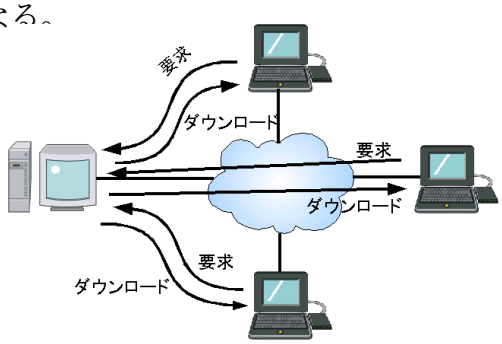

図 2.5: WWW モデル

2.2.7 TCP/IP 通信

TCP/IP で通信するときのアプリケーション層から物理媒体までのデータと処理の流れを記述する。各層では送信されるデータにヘッダと呼ばれる情報が付加される。ヘッダはその層で必要とされる情報が組み込まれており送信や宛先の情報、プロトコルが運んでいるデータに関する情報が含まれている。下位層から見ると上位層から受け取るものは全て単なる 1 つのデータとして認識される。TCP/IP による通信の例として文字列を 2 つのコンピュータ間でやり取りする場合を挙げる。

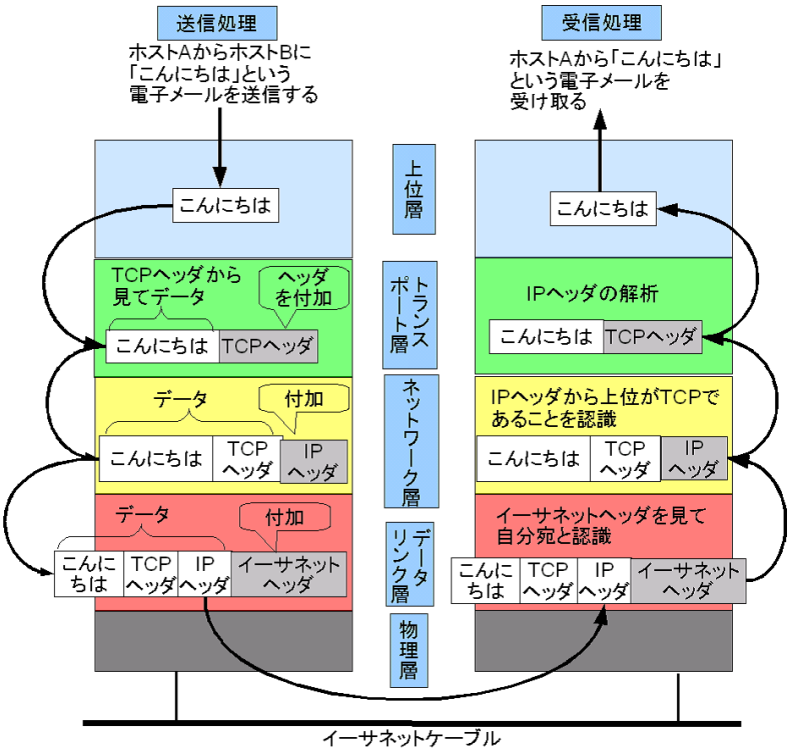

図 2.6: 階層モデルによる通信

送信処理

- アプリケーションの処理
  アプリケーションプログラムでは OSI 参照モデルのプレゼンテーション層に該当する符号化の処理が行われる。メールを送信する際に TCP にコネクションの確立を指示し、TCP のコネクションが確立されたらそれを利用してデータを送るアプリケーションのデータは下位層の TCP に渡され実際の転送処理が行われる。

- TCP モジュールの処理
  TCP はアプリケーションの指示によりコネクションを確立したり、データを確実に相手に届けるために信頼性のあるデータ転送を提供する。機能を実現させるためアプリケーションデータの前に TCP のヘッダが付けられる。TCP

のヘッダには送信ホストと受信ホストのアプリケーションを識別するためのポート番号、そのパケットのデータが何バイト目のデータなのかを示すシーケンス番号、データが壊れていないことを保障するチェックサムなどが含まれる。そして、TCP ヘッダを付けたデータを IP に送る。

- IP モジュールの処理
  IP では TCP から渡された TCP ヘッダとデータのかたまりを 1 つのデータとして扱う。TCP ヘッダの前に宛先の IP アドレスや送信元の IP アドレス、IP ヘッダの次に続くデータが TCP か UDP 可であるかを示す情報などが含まれる。IP パケットが完成したらパケットを送信すべきルーターやホストを決定し、ネットワークインターフェースのドライバに IP パケットを渡して、実際に送信処理を行わせる。通信先の機器の MAC アドレスが不明な場合は、ARP(Address Resolution Protcol) を利用して MAC アドレスを調べイーサネットドライバへ MAC アドレスとデータを渡して送信処理を行わせる。

- ネットワークインターフェースの処理
  IP から渡された IP パケットはイーサネットドライバから見ると単なるデータであり、このデータにイーサネットヘッダを付け送信処理を行う。イーサネットヘッダには宛先の MAC アドレスと送信者の MAC アドレス、イーサネットヘッダに続くデータのプロトコルを示すイーサネットタイプが書き込まれる。異常の処理を行った後に物理層により相手先に運ばれる送信中に FCS(Frame Check Sequence) がハードウェアで計算され、フレームの最後に付けられる。FCS はノイズなどによりデータが破壊されたことを検知するためのものである。

受信処理

受信ホストは送信ホストとまったく逆の受信処理を行う。下の図は階層化によるパケットの構造である。

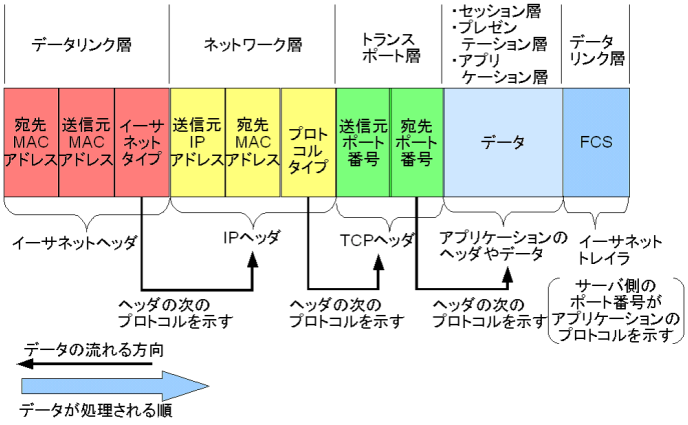

図 2.7: パケット構造

- ネットワークインターフェースの処理
  データを受けとったホストは、まずイーサネットヘッダの宛先 MAC アドレスが自分宛のものか調べ自分宛でない場合にはそのフレームは捨てられる。次にイーサネットタイプフィールドを調べ、イーサネットプロトコルが運んでいるデータを調べる。本件では IP であるので IP を処理するルーチンにデータを渡す。ARP などほかのプロトコルの場合にはプロトコルに合った各ルーチンにデータを渡すことになる。また、処理できないプロトコルの値がイーサネットタイプフィールドに入っている場合はデータを捨てる。

- IP モジュールの処理
  IP のルーチンに IP ヘッダ以降のデータが渡されるとそのまま IP ヘッダを処理する。送信 IP アドレスが自分のホストの IP アドレスであれば受信する。ルーターの場合は受信する IP パケットの宛先は殆どが自分宛ではないため経路制御表から次に送るホストやルーターを調べて転送処理を行う。自分宛の IP パケットの場合は上位プロトコルを調べる。そして、TCP の場合なら TCP ルーチンに、UDP の場合であれば UDP の処理ルーチンにデータを送る。

- TCP モジュールの処理
  TCP ではチェックサムを計算してデータが壊れていないかを確認した後、データを順番どおりに受信しえいるかどうかを確認する。また、ポート番号を調べて通信を行っているアプリケーションを特定する。データがきちんと届いた場合には送信ホストにデータが届いたことを確認するために「確認応答」を返す。この確認応答がデータを送信したホストに届かない場合には、送信ホストは確認応答が届くまでデータを繰り返し送信する。受信ホストはデータを正しく受信した場合にはポート番号で識別したアプリケーションプログラムにデータを渡す。

- アプリケーションの処理
  受信側のアプリケーションは送信側が送信したデータをそのまま受信することになる。受信したデータを解析しホスト B 宛のメールであることを知る。受信したらハードディスクにメッセージを格納する。全ての電子メールのメッセージを格納できたら、処理が正常に終了したことを送信元のアプリケーションに伝える。途中でメッセージを格納できなかった場合は異常終了のメッセージを送信する。以上全ての階層の処理を経て、はじめてディスプレイ上に「こんにちは」と表示させる事が可能になる。

プロトコルのパケットはプロトコルが利用するヘッダと、そのプロトコルの上位層が利用するデータから構成されている。ヘッダの構造には細かい部分まで明確に仕様が決められており、上位プロトコルを識別するフィールドの位置やビット数、チェックサム計算法やどこのフィールドに入れるかなど多岐にわたっている。

通信する双方のコンピュータでプロトコルの識別番号やチェックサムの計算方法が違えば通信は不可能になる。このように、パケットのヘッダはプロトコルの仕様が目に見える形で存在しておりプロトコルの顔といえる。

2.2.8 TELNET

TELNET[5][6][12][13] とは遠隔ログインを行うためのものであり、遠く離れたコンピュータにログインして、そのコンピュータでプログラムを実行させることを可能にする機能である。遠隔ログインは TSS(Time Sharing System) のような環境を実現するアプリケーションであり、メインフレームと端末の関係をコンピュータネットワークに対応したものといえる。TSS では中央に処理能力の高いコンピュータが存在し、そのコンピュータが処理能力を持たない複数の端末が端末専用の通信回線で接続されていた。このような関係を自分の使用しているコンピュータとネットワークの先に接続されているコンピュータの間で実現したものか TELNET プロトコルである。TELNET を利用することでネットワーク上にある全てのコンピュータにログインすることが可能でありそれにはログインするしたいコンピュータに自分のログイン名とパスワードが登録されている必要がある。TELNET による通信は、自分のコンピュータとネットワーク上の全てのコンピュータ間に端末用の通信回線を仮想的に敷設したようにイメージすることができる。このように考えるとパケットネットワークの応用性の広さが分かる。
TELNET は 1 つの TCP コネクションを利用する。この通信路を通して相手のコンピュータにコマンドが文字列として送信され、相手のコンピュータで実行される。これは、自分のキーボードとディスプレイが相手のコンピュータの内部で動作している「シェル」に接続されているイメージである。シェルとはキーボードやマウスからユーザのコマンドを解釈して、それを OS に実行させるインターフェースのプログラムである。

TELNET のサービスはネットワーク仮想端末の機能とオプションのやり取りをする 2 つの基本サービスに分けることができる。

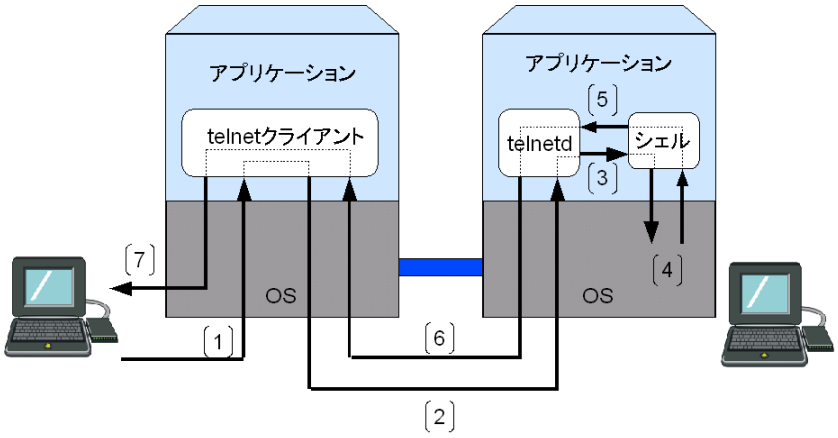

図 2.8: TELNET

1. キーボードから文字列が入力される
2. 行モード、透過モードなどのモード処理をして telnetd へ (1) の文字列を送信する
3. シェルにコマンド列を送信する (厳密には OS を経由する)
4. シェルからコマンド解釈してプログラムを実行、結果を得る
5. シェルからコマンドの出力を受け取る (厳密には OS を経由する)
6. 行モード、透過モードなどのモード処理をして telnet クライアントに送信する
7. NVT の設定に従い画面へ出力する

ネットワーク仮想端末 (NVT: Network Virtual Terminal)

ネットワーク仮想端末はどのような種類のコンピュータの組み合わせでも画面表示がおかしくならないようにする機能であり、OSI 参照モデルのプレゼンテーション層に該当する機能である。TELNET では NVT というネットワーク標準の端末を定義し、その端末に合わせて画面描写の制御を行う。端末により画面の文字コードや画面を消去したりスクロールさせたりするコードが異なる。これを、TELNET を使用するときには NVT に従わせることでどのようなコンピュータ間でも画面が乱れることなくログインして作業をすることができる。NVT を定義したことにより、異なる何百種類の端末間で通信をしても、今後登場する新たな端末でも動作を保障することができる。

オプション

TELNET ではユーザが入力した文字以外にもオプションをやり取り機能が用意されている。NVT を実現するを実現するための画面制御情報はこのオプションの機能を使用して送信される。また、TELNET には行モードと透過モードの 2 つモードがあり、これらの設定は TELNET クライアントと TELNET サーバの間でオプション機能を利用して設定される。

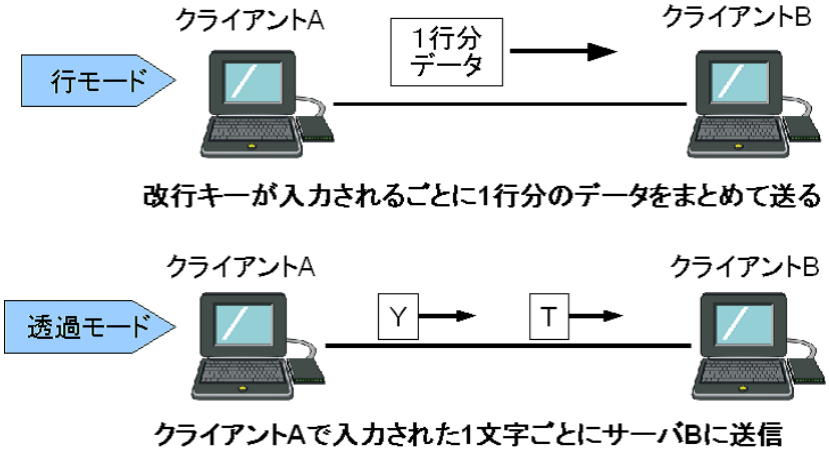

図 2.9: 行モードと透過モード

2.3 データリンク

データリンク層 [5][6] のプロトコルは、通信媒体で直接接続された通信間で通信するための仕様を定めています。通信媒体には同軸ケーブル、ツイストペアケーブル、光ファイバー、電波、赤外線など多種類であり、機器間をリピーター、ブリッジ、ハブなどの機器が中継する場合もある。実際に機器間で通信を行う場合には OSI 階層モデルでいう物理層とデータリンク層がともに必要になる。物理層は 2 進数の 0 と 1 を電圧の変化や光の点滅に変換する働きを持つのに対しデータリンク層では 0 と 1 の数字の列を「フレーム」に分けて意味のある塊として相手の機器に伝えます。このデータリンクはネットワークの最小単位と言ってもよい、世界中を結ぶインターネットによる通信も細かく見るとデータリンクの集合といえる。

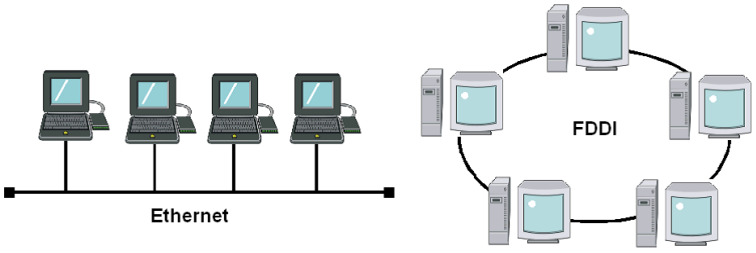

図 2.10: Ethernet・FDDI

2.3.1 コネクション型・コネクションレス型

ネットワークでデータの転送にはコネクション型とコネクションレス型がある。コネクション型としてはATM、フレームリレー、TCPなどのプロトコルがあり、コネクションレス型イーサネット、IP、UDPなどのプロトコルがある。

- コネクション型
  コネクション型はデータの送信を開始する前に送信ホストと受信ホスト間で回線を接続する。これは、プロトコルの階層によって少し意味が変わるがデータリンクの場合には物理的な回線の接続と考えてよい。電話の通信のように相手の電話番号を入力して相手が電話に出てから話をすることとほぼ同じと考えられ、通信の前後にコネクションの確立と切断の処理を行う必要があるが、相手が通信不可能な場合には無駄なパケットを送らずに済む。

- コネクションレス型
  コネクションレス型はコネクションの確立や切断処理がなく、送信したいコンピュータはいつでもデータを送信することができる。逆に受け取る側は、いつ誰からデータを受信するか分からないのでデータを受け取っていないかどうかを常に確認する必要がある。また、通信相手がいるかどうかの確認は行われないため受信相手がいない場合や相手に届かない場合もデータを送信することができる。コネクションレス型は機器にとってはとても有効な手段といえる。手続きや規定を省略することで処理を単純化させることが可能となり、低コストの製品開発や処理の負担を軽減することが可能である。

2.3.2 Ethernet

現在もっとも普及しているデータリンクである。ほかのデータリンクに比べると制御の仕組みが単純なためネットワークインターフェースがデバイスドライバが作りやすいという特徴があるため、現在では安い価格で販売されている。低価格化は普及に非常に大きな役割を果たした。さらに、100Mbps、1Gbpsと高速ネットワークへの対応が進み、互換性と将来性を備えたデータリンクとなっている。もともとは米国のXerox社とDEC社が考案した通信方式で、このときEthernet[5][6][7]と命名された。その後イーサネットはIEEE802.3委員会によって規格化されたが、両者のイーサネットにはフレームのフォーマットに違いがあるためIEEE802.3仕様のイーサネットのことを802.3 Ethernetということがある。

イーサネットには通信ケーブルや通信速度が違うたくさんの種類がある。10BASEの”10”、100BASEの”100”、1000BASEの”1000”はそれぞれ10Mbps、100Mbps、1Gbpsの伝送速度を意味している。後ろにつく”5”、”2”、”T”、”F” などの文字は媒体の違いを示している。通信速度が同じで通信ケーブルが違う場合にはそれぞれの通信媒体を変換できるリピータやハブなどで接続することができ、通信速度

が違う場合には速度変換機能を持つブリッジやスイッチングハブでなければ相互に接続することはできない。尚、本研究では 10BASE-T を利用している。

イーサネットの種類	ケーブルの長さ	最大ノード数	ケーブル種類
10BASE2	185m	30	同軸
10BASE5	500m	100	同軸
10BASE-T	100m	-	ツイストペアケーブル
10BASE-F	1000m	2	光ファイバー (MMF)
100BASE-TX	100m	-	ツイストペアケーブル
100BASE-FX	412m	2	光ファイバー (MMF)
100BASE-T4	100m	-	ツイストペアケーブル
1000BASE-CX	25m	-	シールドされた銅線
1000BASE-SX	220〜550m	2	光ファイバー (MMF)
1000BASE-LX	550m/5000m	2	光ファイバー (MMF/SMF)
1000BASE-T	100m	-	ツイストペアケーブル

表 2.1: イーサネットの種類と特徴

2.3.3 イーサネットの CSMA/CD 方式

イーサネットはバス型ネットワークを基本とし、CMSA/CD 方式でデータの送信制御をしており次のような機能がある。

- 搬送波が流れていなければ (データが流れていなければ) 全てのステーションはデータを送信しても良い
- 衝突が発生したかどうか検出し、衝突が発生した場合には送信をやり直す。

仕組みを図を用いて下記する。

1. 誰も送信していないことを確認する
2. データを送信する
3. データを送信しながら
4. 電圧を監視する。

- 送信終了まで電圧が規定範囲内であったら、正常にデータが送信できたと判断する
- 送信途中で電圧が規定値の範囲外になった場合には衝突が発生したと診断する
- 衝突が発生した場合には送信を停止し、乱数で発生させた時間の間を待ってから再度データを送信する。

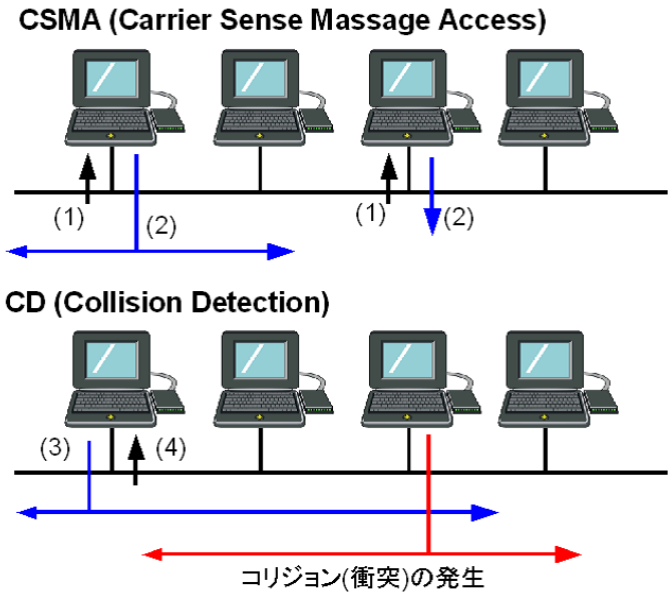

図 2.11: CSMA/CD 方式

2.3.4 フレームフォーマット

イーサネットのヘッダには宛先 MAC アドレスのフィールドが 6 オクテッド送信元 MAC アドレスのフィールドが 2 オクテッド、そしてタイプと呼ばれるフィールドが 2 オクテッドの合計 14 オクテッドからなるヘッダで構成されており、最後に FCS(Frame Check Seaquence)[5][6] という 4 オクテッドのフィールドがある。

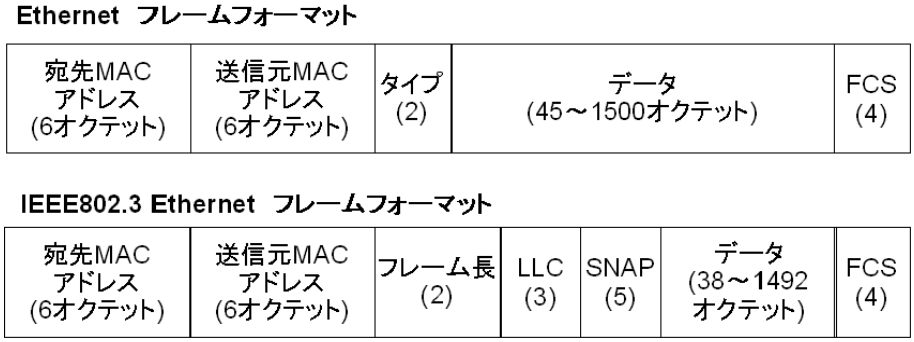

図 2.12: フレームフォーマット

宛先 MAC アドレスには宛先のノードの MAC アドレスが格納される。送信元 MAC アドレスにはイーサネットフレームを作り出した送信元ノードの MAC アドレスが格納される。タイプにはデータ部で運んでいるプロトコルを表す番号が格納され、イーサネットの上位層のプロトコルを示している。データ部の先頭からはタイプで示されたプロトコルのヘッダやデータが格納される。

最後の FCS はフレームが壊れていないかどうかをチェックするフィールドである。通信中に電気的なノイズが発生すると送信したデータがビット化けを起こしデータが壊れる可能性があるり、この FCS の値をチェックすることでノイズによるエラーフレームを破棄することができる。FCS はフレーム全体を特定のビット列で排他的論理和による割り算を行い、その余りを格納する。受信側でも同じ計算を行い FCS の値が同じになったらデータが正しく届いたと判断する。

IEEE802.3 Ethernet は通常のイーサネットとはフォーマットが異なる。タイプの部分が長さになり、その後に LLC と SNAP というフィールドが続く。この SNAP の中にプロトコルを表すタイプのフィールドがあり、SNAP タイプ値はイーサネットとほぼ同じである。

2.4 SPI 接続

SPI(Serial Peripheral Interface)[1] はコンピュータ内部で使われるデバイス同士を接続するバスである。パラレルバスに比べて接続端子数が少なくてすむシリアルバスの一種で比較的低速なデータ転送を行うデバイスに利用される。
従来のデータバス、アドレスバス、制御バスには少なくとも十数本の信号を接続する必要があった。メインメモリなど高速に接続する必要があるデバイスを除いて、それほど速度を必要とされないデバイスに関しては IC パッケージも小型化できることから、省ピンで接続できる形態が望まれた。このような背景から、いくつかのシリアルバス規格が提唱された。(SPI、I2C、MicroWire など) 信号線は 4 本で構成され、1 つのデバイスを接続する場合は SS を固定することで 3 本で接続できる。

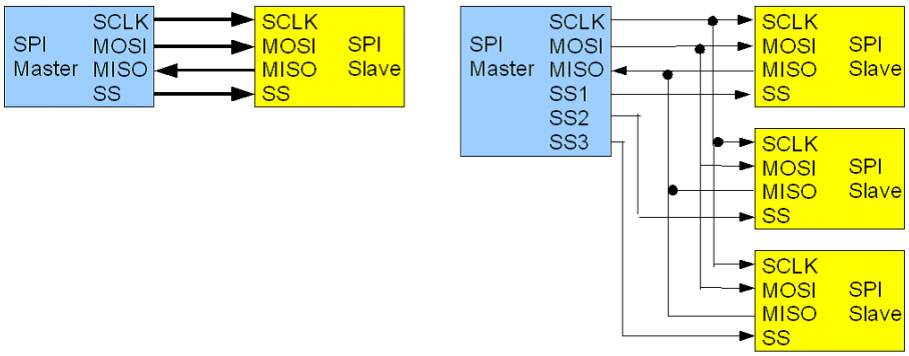

図 2.13: SPI バスの接続

2.5 組み込みシステム・ISP

組み込みシステム

組み込みシステム (Embedded system)[2][3][4] とは特定の機能を実現する目的でコンピュータを組み込んだ電子機器の総称であり、パーソナルコンピュータ等の汎用的なシステムと対比される。

組み込みシステムが発達する前は、電子制御を行う仕組みをアナログ回路やデジタル回路といったハードウェアによる回路により構成していたが、回路を変更する必要があるためコストがかかるといる問題があった。1980 年年代以降コンピュータ (特にマイクロコントローラ) の発達により、コンピュータを用いた制御方式にすることで、回路は変更せずソフトウェアのみを変更すれば機能の追加が可能になり機能追加に必要なコストが削減された。このため殆んどの電化製品は組み込みシステムを用いて付加価値として新機能を追加するようになっている。他にも、工場などでの作業を自動化する産業ロボットや工作機械などの産業機器もコンピュータを内蔵した組み込みシステムである。

特徴

- 汎用コンピュータに対して非常に数と種類が多い
- ソフトウェアだけでなくハードウェアも専用のものを開発することが多く、ハードウェアに対応したデバイスドライバを作る必要がある
- 大量生産される製品の場合にはコストが非常に重要であるため、必要最低限のメモリと安価な CPU で動作する 1 チップマイコンが利用されることが多い
- 組み込みシステムではユーザがプログラムを入れ替えたり更新することは想定されないため、汎用コンピュータよりも自由にシステム構成が可能
- 特に低資源の環境では OS を使用しないことが多い
- ソフトウェアは C 言語で記述されることが多い。32 ビット以上のマイコンなど比較的ハードウェアの資源が豊富な環境では C++や Java が使用されることもある
- 携帯電話、デジタル家電、自動車など必要とする機能が多岐にわたるシステムは、複数のマイコンなどを組み合わせたものであり、開発には数年規模を必要とする場あるため、大規模組み込みシステムと呼ばれることもある

ISP(In-System Programming)

ISP[2][3][4] はある種のマイクロコントローラやその他プログラム可能な電子部品において事前にプログラムを書き込んでからシステムに組み込むのではなく、組み込み済みの状態でプログラムを書き込むことである。この機能の利点はシステムの組み立て前に書き込み段階を別途に設ける必要がなく電子機器の製造者がプログラムの書き込みとテストを 1 つの製造工程で行えることである。書き込み済みチップをメーカーなどから購入する代わりに製造者がシステムの製造ラインでチップへの書き込みができるので製造期間の途中でもコードや設計の変更を可能にする。

一般的に ISP をサポートしたチップは全ての電圧をシステムの通常の供給電圧から作り出す回路を持っており、ライタとはシリアルプロトコルで通信する。

2.6 Atmel AVR

Atmel AVR は Atmel 社が製造している RISC ベースのマイクロコントローラの総称である。PIC マイコン同様に回路構成が簡単で CPU、メモリ (ROM、RAM)、I/O、データ記憶用の EEPROM、クロック発信回路、タイマーなどが 1 つのチップに搭載されており書き込まれたプログラムにより制御される。ISP に対応し、コンパレータを内蔵するなど後発的であったため PIC マイコンに不足する点を補う構成をしている。

C 言語 [14] でのプログラミングを意識しており、アセンブラを含んだ開発環境の「AVR Studio」が無償で提供されている。また、AVR Studio に WinAVR を組み合わせて使用することにより楽に開発環境を整えることができる。

プログラム格納用の ROM は全種類で FlashROM を採用しており、殆んどの命令を 1 クロックで実行するため、MHz あたりの計算量は 1MIPS に達する。

AVR という名前はチップを設計した Alf Egil Bogen と Vegard Wollan の名前と RISC から取られている。

種類は起源となった 90S シリーズと、それを大容量化し I/O を拡張した Mega シリーズ、高機能化・低消費電力化・停電源化した Tiny シリーズがある。本研究では Mega シリーズの Atmega168 を使用している。

2.7 超高輝度 LED

LED(Light Emitting Diode)[1] は順方向に電圧をかけた際に発光する半導体素子でありエレクトロミネセンス効果により発光している。半導体を用いた pn 接合という構造で構成され、発光はこの中で電子の持つエネルギーを直接、光エネルギーに変換するとこで行われる。電極から半導体に注入された電子と成功は異なったエネルギー帯である伝導体と価電子帯を流れ、pn 接合部分付近で金生態を超えて再結合する。再結合の際にバンドギャップに相当するエネルギーが光として放出される。超高輝度 LED とはその名で販売されており、以前の LED よりも輝度が高いものとなっている。

3 提案

2.7 で超高輝度 LED のことを記述したが LED は双方向性の高い素子であり、本研究では超高輝度 LED の発光ではなく超高輝度 LED に光をあてた場合に起こる現象である光起電力効果による高い光起電力に特に注目した。
超高輝度 LED の光起電力は受光する光の環境の変化で電圧に変化が起こる。この特性を利用しセンサとして光の環境の変化を感知し、検知対象の検知、進入経路検知を可能にするセキュリティシステムを構築する。

3.1 μ IP とセンサ

検知対象の検知、進入経路の検知の基本的な構成は超高輝度 LED をセンサとして用いて、その光起電力の電圧を Atmega168 に取得さるアプリケーションプログラムを実装しておき、ホストコンピュータに継続的に電圧の値を送信させ、検知対象者の存在により発生した影により超高輝度 LED の光起電力の電圧に変化があった場合は対象者の検知となるような構成である。
ホストコンピュータへの電圧の送信にはは Atmega168 に μ IP を実装しておき、TCP/IP によって通信できるようにする。
本研究ではセンサと μ IP を実装した Atmega168 を組み合わせた実験装置と実験装置とホストコンピュータ間で通信可能にするためのイーサネットコントローラーを作製し、実際にセンサからの光起電力により発生した電圧の取得し AD 変換したデータの通信を行った。実験については 4 章の実験と考察で詳しく記述する。

開発環境

・AVR Studio 4(統合開発)
・WinAVR(C コンパイラ)
・AVRISP mk　(書き込み用ライタ)

開発の流れ

1. μ IP のソースコード内のアプリケーションを超高輝度 LED の発電する電圧を取得させるプログラムに変更し、コンパイルして生成した HEX ファイルの μ IP のソースコードを AVR ライタ (AVRISP mk　) を用いて Atmega168 に書き込む
2. μ IP を実装した Atmega168 を SPI 接続を用いてイーサネットコントローラと接続し 10BASE-T を構築する。さらに、Atmega168 の PC0 にセンサ (超高輝度 LED) を接続し実験装置を作製
3. ホストコンピュータとルーターを介して接続する。
4. TELNET を使用し、Atmega168 にアプリケーションを実行させる

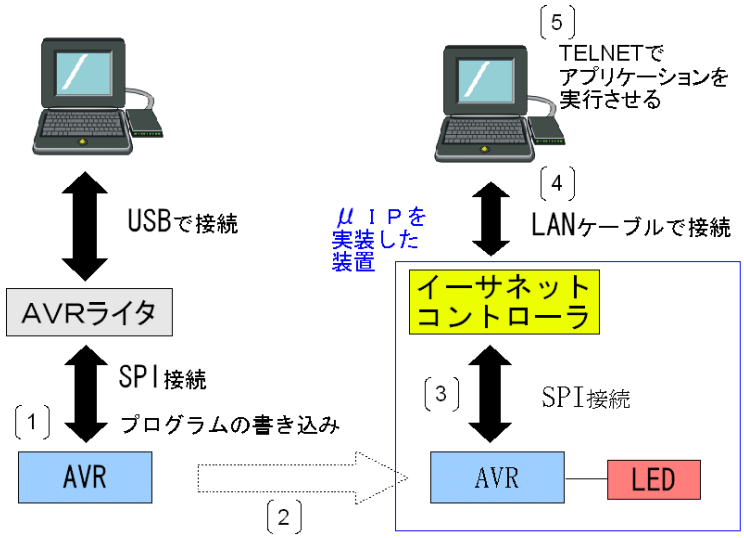

図 3.1: 開発の流れ

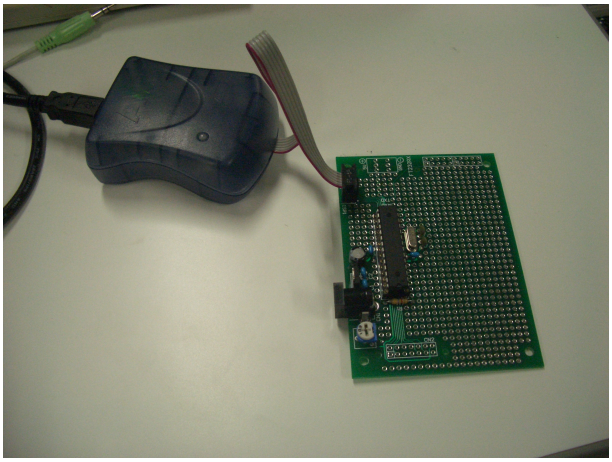

図 3.2: プログラムの書き込み

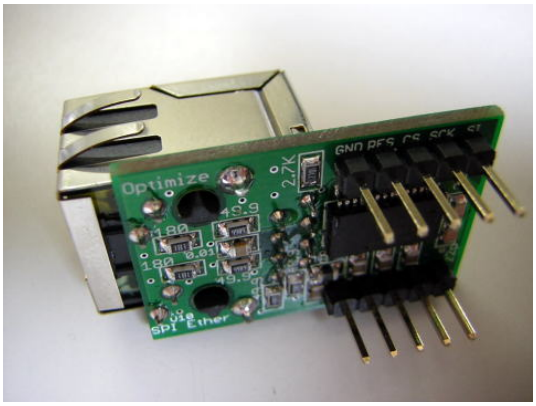

図 3.3: イーサネットコントローラ

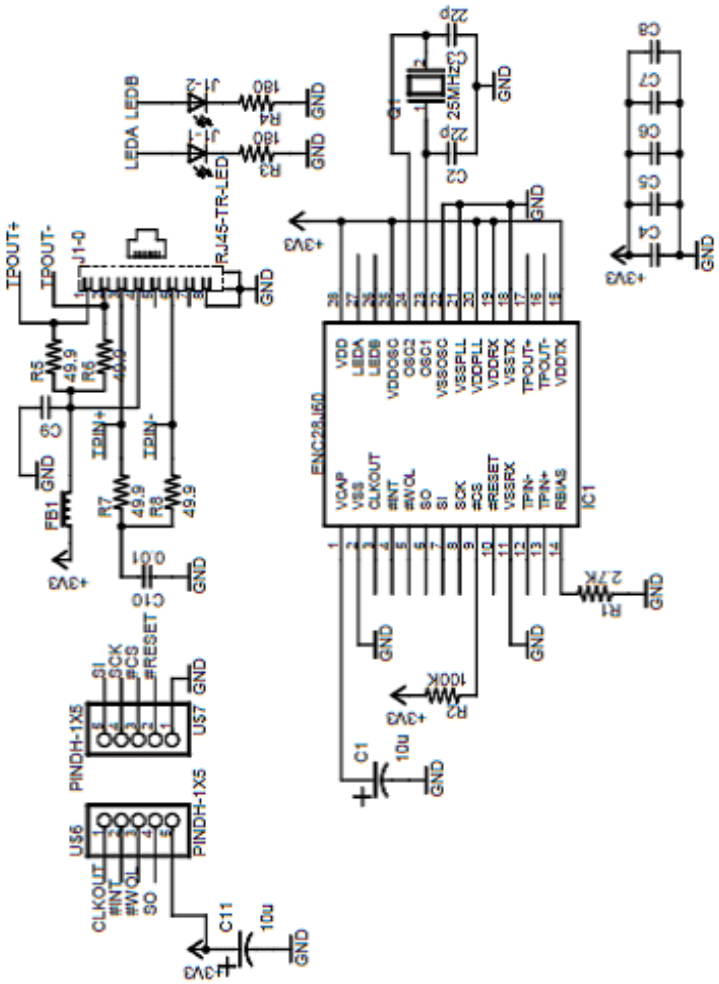

図 3.4: イーサネットコントローラ回路図

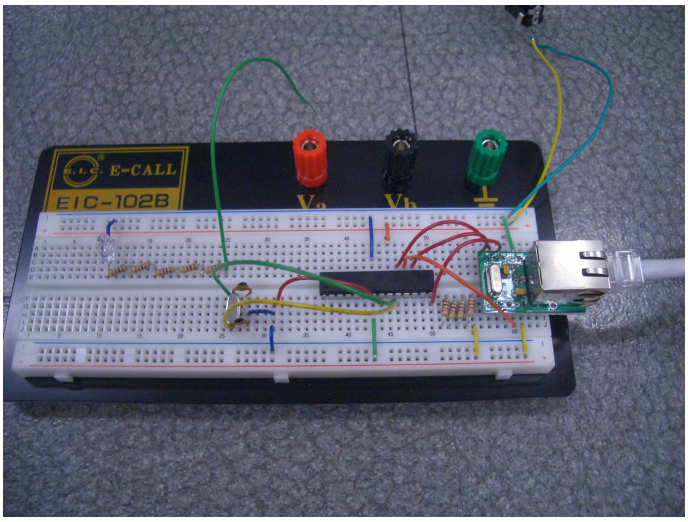

図 3.5: 実験装置

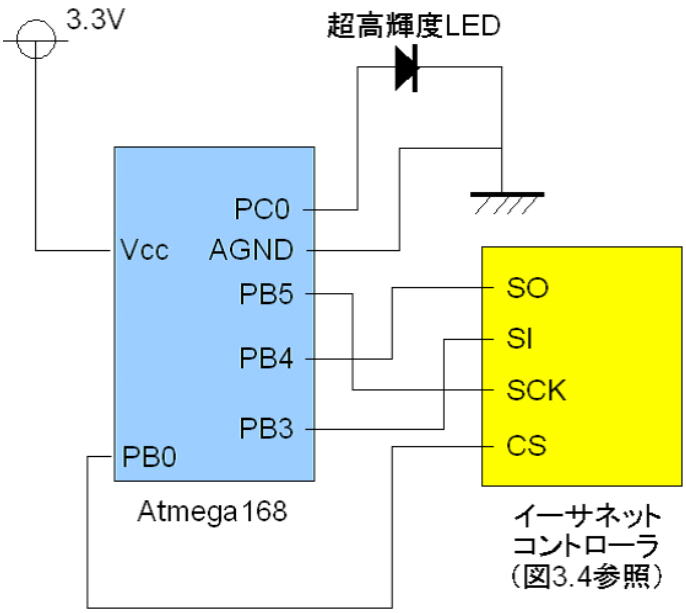

図 3.6: 実験装置回路概要

3.2 センサネット

センサネットは主に検知対象の経路検知を目的として考案したものである。基本的な構成はは 3.1 で記述した実験装置を複数作製し、ルーターでホストコンピュータと接続する構成である。尚、実験装置の IP アドレス、MAC アドレスは実験装置ごとに μ IP の設定ファイルを変更し、再コンパイルした後に各 Atmega168 にプログラムの書き込みを行った。今回は実験装置を 2 つ作製し、2 つの装置間での人の移動を検知する実験を行った。実験については 4. 実験と考察で記述する。

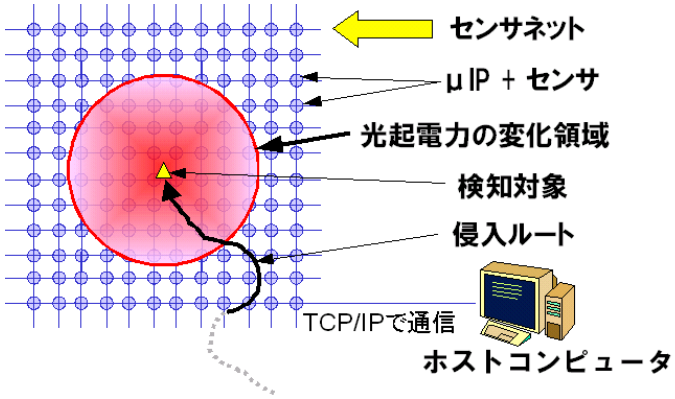

図 3.7: センサネット

ルーターを介しホストコンピュータと接続された装置へホストコンピュータが順番にデータを取得させる命令を送信し、装置はその命令を受信しデータをホストコンピュータへ送信する。送信されてきたデータを元に信号を処理し検知対象者の検知を行う。

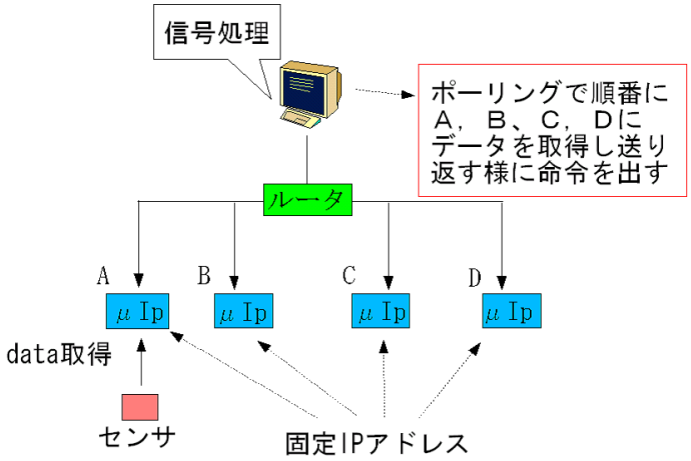

図 3.8: システム構成

3.3 システムの処理

ここではセンサネットシステムの処理の流れについて記述する。

処理の流れ

1. 全ての装置に対し電圧の取得命令を送信する
2. センサの電圧の変化を比較するためにセンサの電圧の初期値を取得する
3. 再び全てのセンサに電圧取得命令を送信する
4. 初期値の電圧と比較し、電圧に変化があればホストコンピュータへ電圧の値を送信する
5. 装置から送信されてきた電圧を取得し環境の変化が起こったか判定する
6. 環境に変化があれば装置の位置を特定し、表示する
7. 再び全てのセンサに電圧取得命令を送信する

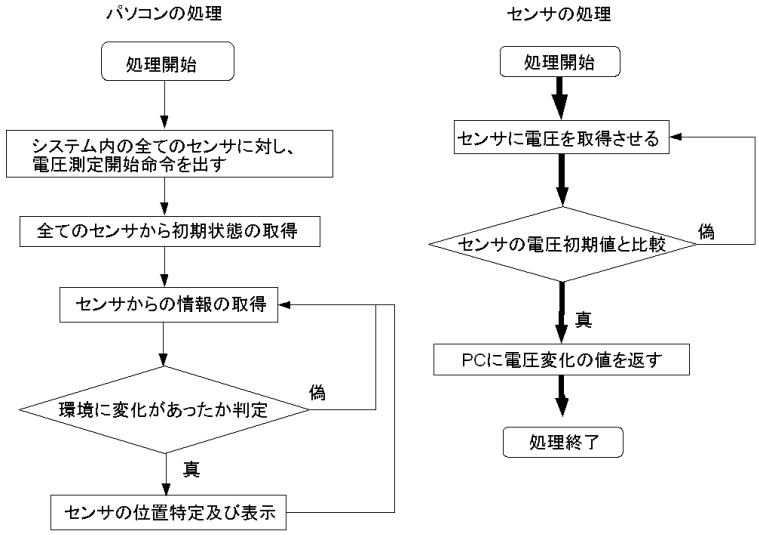

図 3.9: 処理の流れ

4 実験と考察

受光スペクトル

初めに、超高輝度 LED の受光スペクトルの実験を行った。実験方法は光源から分光器を通過して単色となった光を超高輝度 LED に照射し、400〜700nm の波長範囲にわたる光起電力による電圧の変化を測定した。
実験を行った結果、赤色 LED が他の色の LED に比べ受光する光の範囲がもっとも広く、光起電力の電圧も高いことが分かった。その結果を踏まえ本研究の実験装置には赤色超高輝度 LED を使用した。

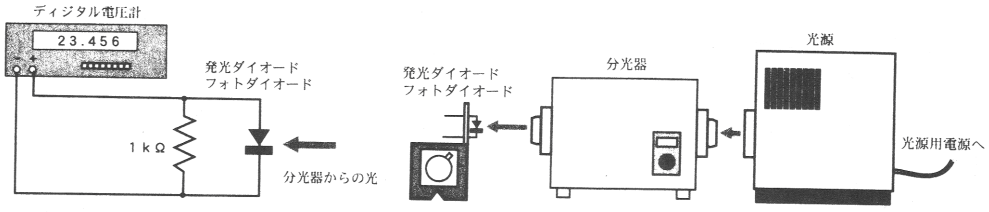

図 4.1: 受光スペクトル実験の機器配置

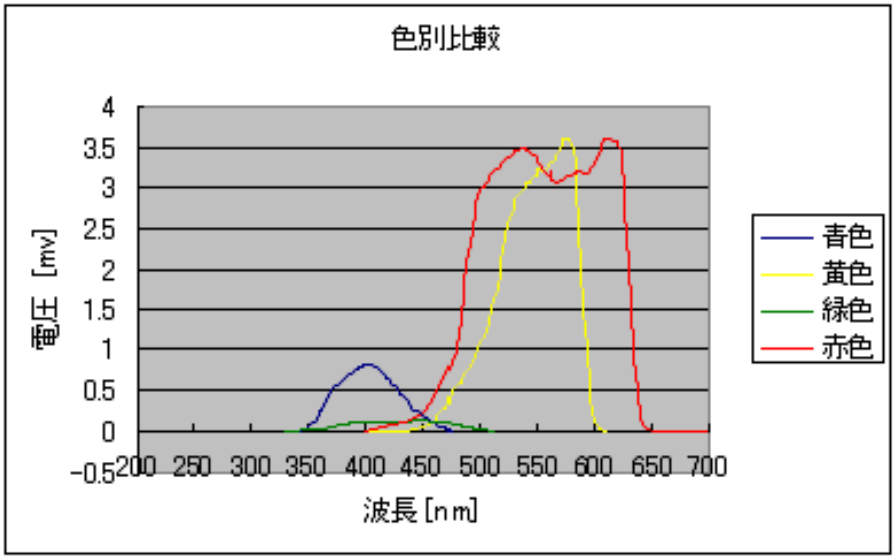

図 4.2: 受光スペクトル

距離に対する電圧特性

次に装置のセンサに光を遮断する物体を近づけていき遮断物とセンサの距離に対する光起電力の電圧の特性を測定した。この実験ではセンサに超高輝度 LED を 1 つ使用した場合と 2 つを直列に接続した場合で行い、各場合のデータを比較した。その結果から LED を 2 つ直列にした場合のほうがセンサとしての感度が高くなることが考えられる。図 4.4、4.5 は遮断物を近づけていないときの電圧取得画面である。データは 8 ビットで構成され電圧取得画面の 255 は 3V に相当する。

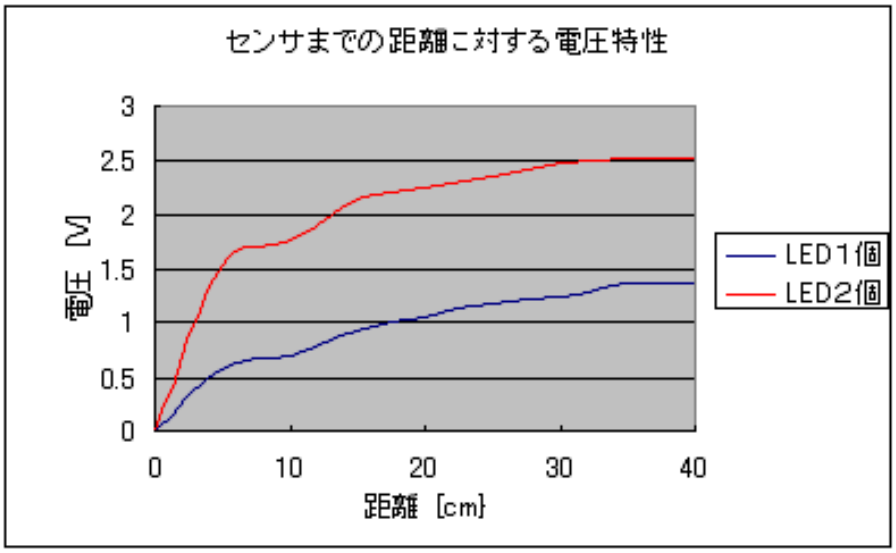

図 4.3: 遮断物の距離に対する電圧特性

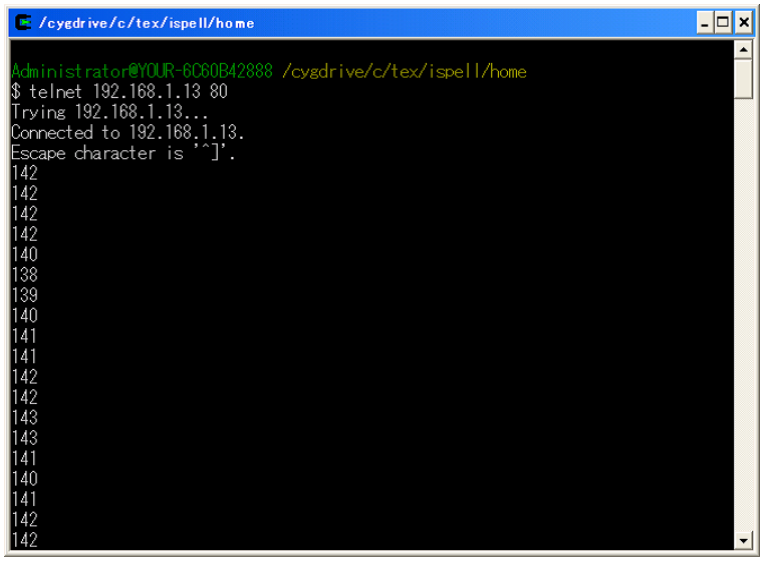

図 4.4: 電圧取得画面 (LED1 個)

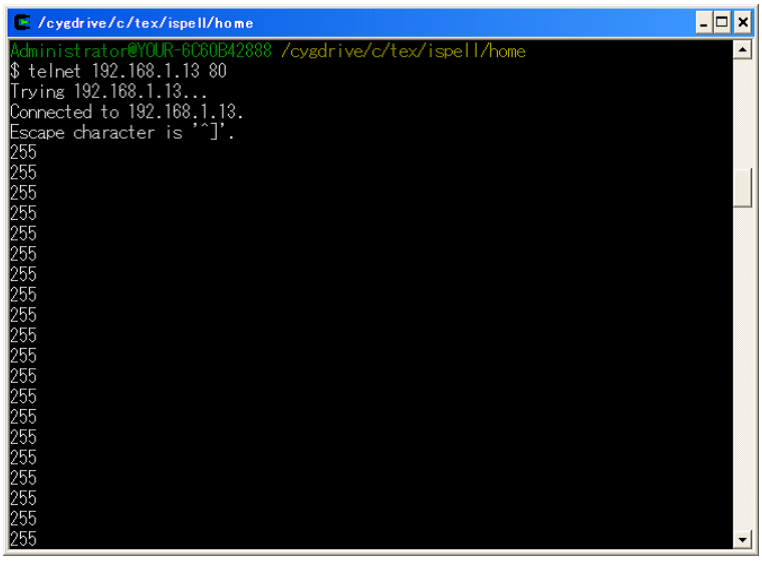

図 4.5: 電圧取得画面 (LED2 個)

また、実験装置を用いてセンサ間での移動の検知の実験と超高輝度 LED とフォトダイオードを同様の装置に搭載した際のセンサとしての感度の比較を行った。

4.1 センサ間での移動検知

実験装置を 2 つ作製し、2 つの実験装置間を人間が移動したときのセンサの電圧の変化を測定し、電圧の変化から 2 つの装置間を人が移動したことを検知する実験を行った。

実験を行った結果、2 つの装置のセンサに時間差で電圧の変化が起こり 2 つの装置間を人が移動したことを検知することができた。電圧取得画面で電圧が 0 になっているところは光の環境の変化を受けており、人を検知したことを表す。

各装置の電圧変化のグラフから同時に処理を開始した装置 1 と装置 2 に時間差で電圧の変化が起きていることが分かる。また、装置を 30 秒間継続してプログラムを実行させた結果、150 回処理を行った。この事から一回の処理にかかる時間は 0.2 秒であることが分かった。

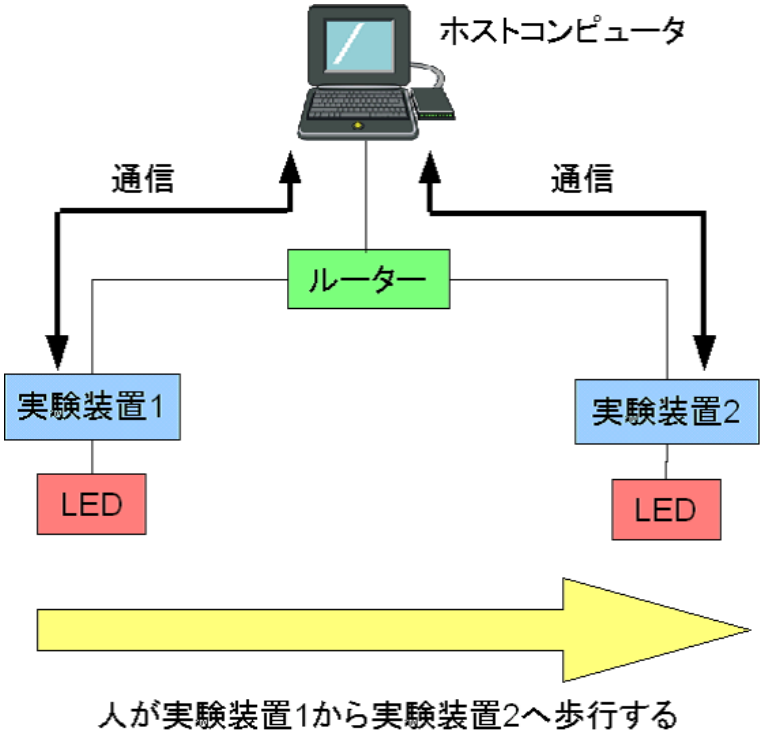

図 4.6: センサ間での移動検知

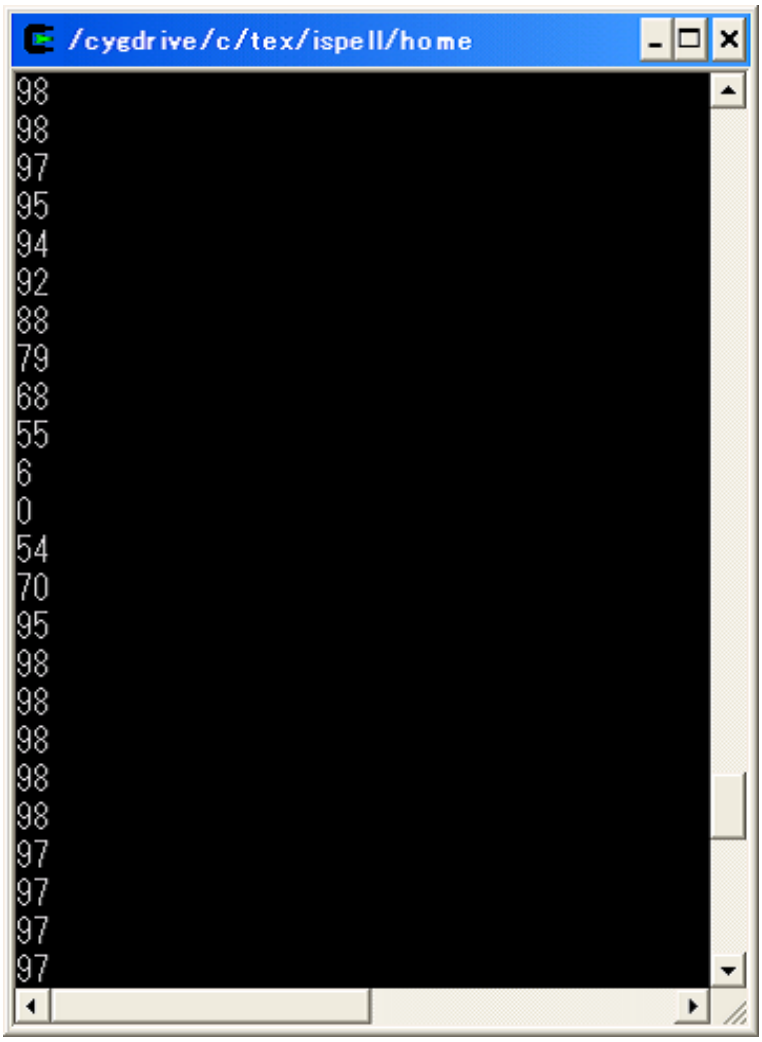

図 4.7: 実験装置 1 の電圧取得画面

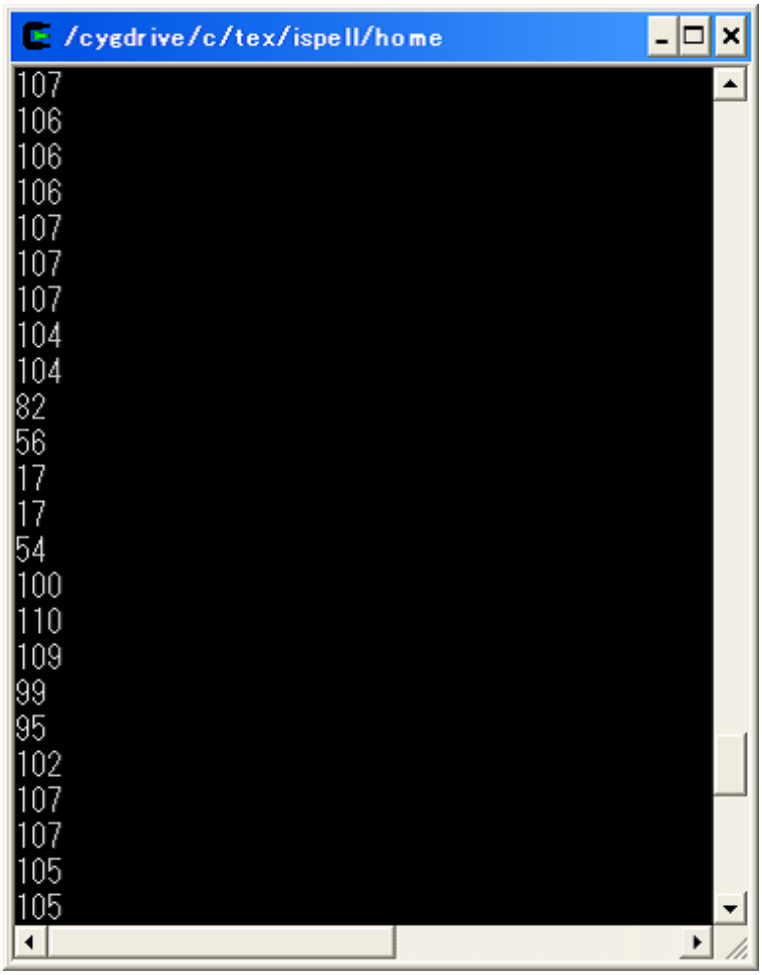

図 4.8: 実験装置 2 の電圧取得画面 (LED1 個)

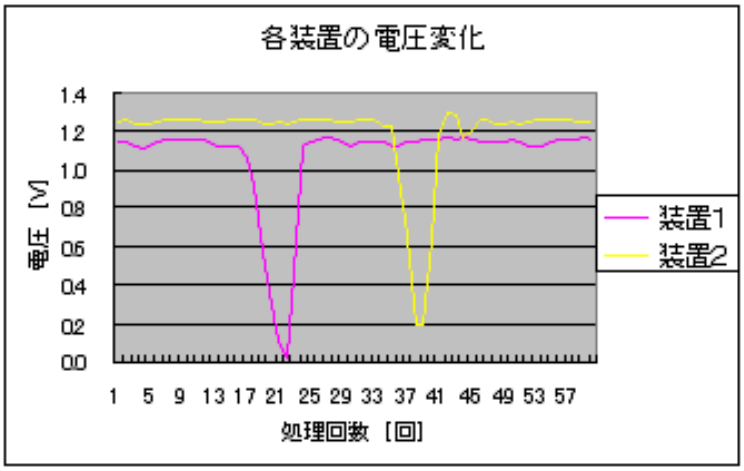

図 4.9: 各装置の電圧変化

4.2 センサの感度比較

センサの感度の比較を行ったところ超高輝度 LED の方がフォトダイオード (Series 5T) よりも発電する電圧が高く、光の環境の変化から検知対象を検知する本システムでは超高輝度 LED の方がセンサとしての感度が高いといえる。

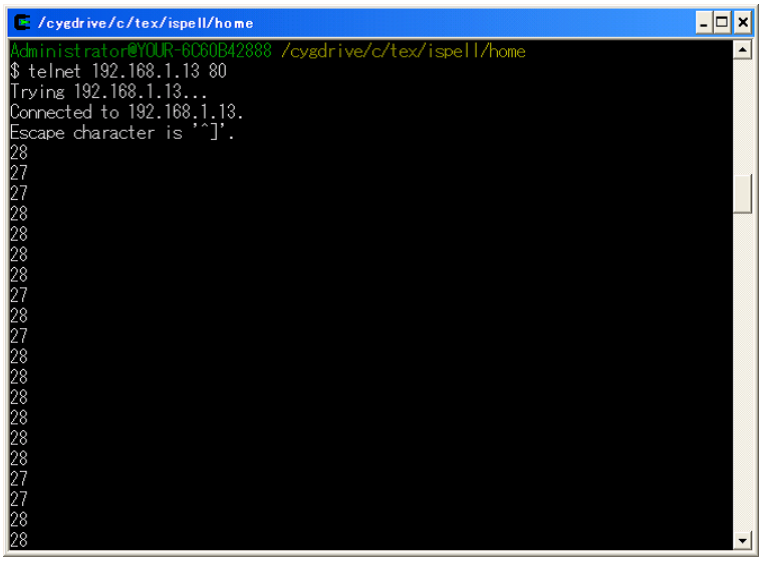

図 4.10: フォトダイオードの電圧取得画面

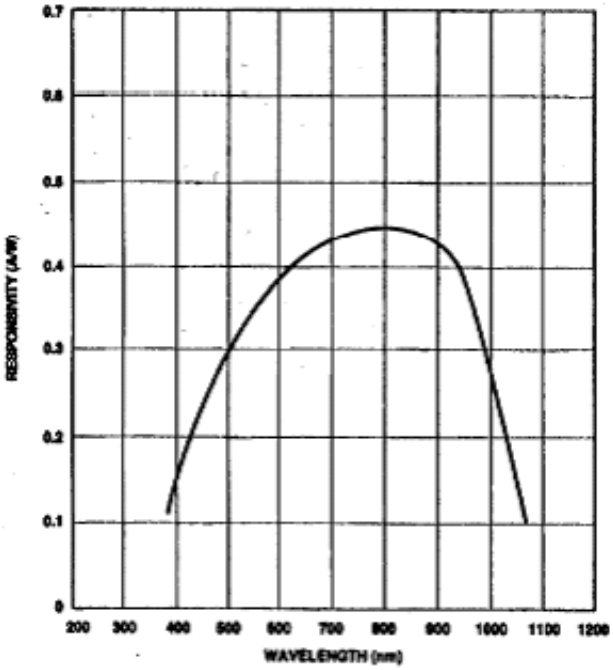

図 4.11: フォトダイオードの受光スペクトル

5 結論

本研究ではアプリケーションプログラムの作製、プログラムの書き込みから実験装置を構築するためのイーサネットモジュール、Atmega168 にセンサとして超高輝度 LED を組み合わせた回路の作製を行った。また、センサネットを用いたセキュリティシステムを構築するために必要な基礎的な実験を行い、その結果よりホストコンピュータへ超高輝度 LED の光起電力の電圧を継続的に送信させ、そのデータから検知対象の検知と 2 つの実験装置間の移動検知が可能であることを証明した。本研究に加え、装置内の処理である取得した電圧を初期値の電圧と比較するプログラムとホストコンピュータの処理である装置から取得した情報を元に光の環境に変化があったかどうかを判定するプログラム、センサの位置特定及び表示のプログラムを作製することにで、センサネットを用いたセキュリティシステムの構築は可能であると考えられる。

実験装置にセンサとして用いる超高輝度 LED を複数個直列に接続することでセンサの感度を向上させることができ、より信頼性のあるセキュリティシステムの構築が可能になるが Atmega168 の動作電圧が 3V であるため Atmega168 は 3V 以上の電圧を測定することはできない。しかし、超高輝度 LED を直列に複数個接続したものを減衰器を通して Atmega168 に接続することでセンサの感度向上が可能になると考えられる。

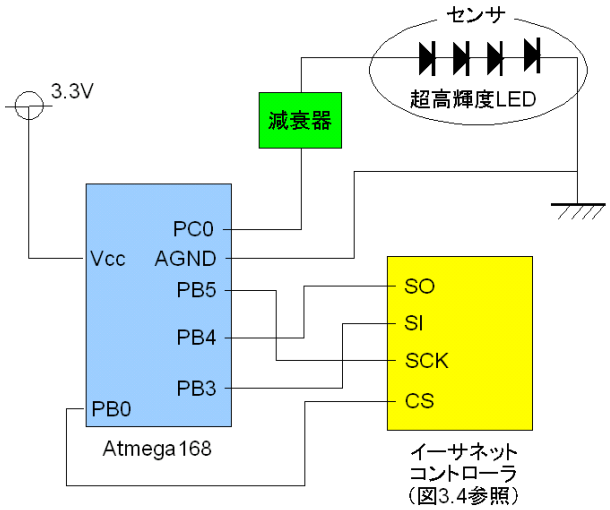

図 5.1: センサ感度向上の例

謝辞

本研究を行うにあたり、終止熱心にご指導して頂いた木下宏揚教授と鈴木一弘氏、ご多忙の折研究室に足を運び様々な面で有益なご助言をして頂いた森住哲也氏に深く感謝いたします。さらに、公私にわたり良き研究生活送らせて頂いた木下研究室の方々に感謝いたします。

参考文献

[1] インターネット・ガジェット設計
武藤 佳恭/著　オーム社

[2] AVR マイコン・リファレンス・ブック
山根 彰/著　CQ 出版

[3] 試しながら学ぶ AVR 入門
土井 滋貴/著　CQ 出版

[4] AVR マイコン活用ブック
松原 拓也/著　電波新聞社

[5] マスタリング TCP/IP 入門編
竹下 隆史・村山 公保・荒井 透・苅田 幸雄/共著　オーム社

[6] マスタリング TCP/IP 応用編
Philip Miller/著　苅田 幸雄/監訳　オーム社

[7] ネットワークはなぜつながるのか
戸根 勤/著 日経 NETWORK/監修 日経 BP 社

[8] optimize SPI イーサネットモジュール基盤
http://optimize.ath.cx/spi_ether/spi_ether.htm

[9] LA Skater
http://www.laskater.com/projects/uipAVR.htm

[10] µ IP Main Page
http://www.sics.se/~adam/uip/index.php/Main_Page

[11] TCP/IP とソケット
南山 智之・佐々木 杉夫/著　共立出版株式会社

[12] Windows ユーザーのための UNIX 入門
藤森 水絵　新紀元社

[13] X ウィンドウによる UNIX 入門
九州工業大学情報科学センター/編　朝倉書店

[14] 改訂新 C 言語入門シニア編
林 晴比古　ソフトバンクパブリッシング

質疑応答

Q; 装置の数を増やすとセキュリティの信頼性が増すのか？
A; 数が多くなるほど信頼性は高くなる。

Q; 実験環境は？
A; 今回は装置は二つのみで、電気のついた部屋で実験を行った。

Q; 赤外線センサの利用は？
A; 自然光などの光がない場合、有効と考えられる。