平成１６年度

卒業論文

題名：無線 LAN を使用した DGPS 測位のリアルタイム化

課程名：商船システム工学課程　航海学コース

学籍番号：２００１５４８

氏名：SAHA KRISHNA KAMAL

指導教官：安田　明生　教授

2005 年 01 月 31 日提出

ABSTRACT

Development of the Real-Time DGPS Calculation System Using Wireless LAN

The Differencial GPS techniques, such as DGPS and RTK-GPS are indispensable when highly accurate service is needed, although the accuracy of GPS positioning has been improved substantially by turning off the Selective Availability which was the largest source of error for GPS receivers till 2000.For the DGPS and RTK-GPS positioning system, the most important task is how to transmit the correction data from the base or reference station to the rover station.

In this research, I used Wireless LAN to transmit GPS data from the reference station to the PC with wireless LAN system using a serial to wireless LAN converter named “EZL-300W” which is connected to the reference GPS receiver with a serial RS232C cross cable. The PC for DGPS calculation in the rover station is connected to the rover GPS receiver with serial RS232C cross cable through the PC’s COM port. I used the serial communication functions of WIN32 API for using the PC’s COM port to collect data from the receivers. In the programming, I mainly used the dialog based MFC of VISUAL C++ programming language. But for the position calculation I used the C/C++ based programs of the lab.

I also performed some experiments inside our campus to find out the range of wireless LAN. To transmit data from the EZL-300W, I used 11Mbps wireless LAN PC card adaptor of IEEE 802.11b standard that uses 2.4GHz.

目次

第１章　序論

1.1　はじめに　・・・・・・・・・・・・・・・・・・・・・・・・・・・・1
1.2　本研究の目的　・・・・・・・・・・・・・・・・・・・・・・・・・・2
1.3　GPS の概要　・・・・・・・・・・・・・・・・・・・・・・・・・・3
1.4　DGPS の概要・・・・・・・・・・・・・・・・・・・・・・・・・・・4

第２章　測位原理及び測位計算

2.1　GPS 単独測位原理・・・・・・・・・・・・・・・・・・・・・・・・ 6
2.2　単独測位計算・・・・・・・・・・・・・・・・・・・・・・・・・・8
2.3　DGPS 測位原理・・・・・・・・・・・・・・・・・・・・・・・・・１０
2.3.1　測位位置補正方式　・・・・・・・・・・・・・・・・・・・・・１１
2.3.2　擬似距離補正方式　・・・・・・・・・・・・・・・・・・・・・１１

第３章　システムの構成

3.1　現在、研究室で使われているシステム・・・・・・・・・・・・・・１２
3.2　システムのリアルタイム化　・・・・・・・・・・・・・・・・・・１３
3.3　通常及び仮想 COM ポート　・・・・・・・・・・・・・・・・・・・１４
3.3.1　COM ポート　・・・・・・・・・・・・・・・・・・・・・・・・１４
3.3.2　通常 COM ポート　・・・・・・・・・・・・・・・・・・・・・・１４
3.3.3　仮想 COM ポート　・・・・・・・・・・・・・・・・・・・・・・１６

第４章　通信及び測位プログラム

4.1　プログラム言語の概要・・・・・・・・・・・・・・・・・・・・・・２０
4.2　通信プログラムについて　・・・・・・・・・・・・・・・・・・・・２５
4.3　測位計算プログラムについて　・・・・・・・・・・・・・・・・・・３２
4.3.1 外部変数　・・・・・・・・・・・・・・・・・・・・・・・・・・３３
4.3.2 データの読み込み　・・・・・・・・・・・・・・・・・・・・・・３３
4.3.3 衛星の選択　・・・・・・・・・・・・・・・・・・・・・・・・・３３
4.3.4 測位計算　・・・・・・・・・・・・・・・・・・・・・・・・・・３３
4.4　プログラムの使い方及び全体の　フローチャート・・・・・・・・３５
4.4.1 プログラムの使い方　・・・・・・・・・・・・・・・・・・・・・３５
4.4.2 プログラム全体のフローチャート　・・・・・・・・・・・・・・３６

第５章　無線 LAN を使った実験及びその結果

5.1　無線 LAN について　・・・・・・・・・・・・・・・・・・・・・・・３８
5.1.1 無線 LAｎの概要　・・・・・・・・・・・・・・・・・・・・・３８
5.1.2 無線 LAN の規格について　・・・・・・・・・・・・・・・３９
5.1.3 無線 LAN（IEEE802.11b）によりネットワーク構成　・・・４０
5.1.4 シリアルインターフェース/無線 LAN コンバータ
（EZL-300W）・・・・・・・・・・・・・・・・・・・・４２
5.1.5 無線 LAN の伝達距離とパフォーマンス・・・・・・・・・　・４７

5.2　無線 LAN の届く場所及び距離の実験結果・・・・・・・・・・・・４８

第６章　DGPS 測位実験及びシステムの評価

6.1　無線 LAN を使った DGPS 測位実験・・・・・・・・・・・・・・・　５１
6.2　実験結果よりシステムの評価・・・・・・・・・・・・・・・・・・５２

第７章　　結論

7.1　まとめ・・・・・・・・・・・・・・・・・・・・・・・・・・・・５６
7.2　今後の課題・・・・・・・・・・・・・・・・・・・・・・・・・・５６

参考文献　・・・・・・・・・・・・・・・・・・・・・・・・・・・・・57

謝辞　・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・58

第１章　序論

1.1　はじめに

GPS は Global Positioning System(全地球測位システム)の略で 1970 年代に米国国防総省（DOD:Department Of Defence）により開発が始められた人工衛星による測位システムである。1993 年末に正式的にシステムとして完成宣言が行われ、現在では 24 衛星の運用で全地球をカバーし、地球上どこでもいつでも高精度の 3 次元測位を可能とする測位システムとして世界的に広く受け入れられている。もともとの主目的は米国の他、北大西洋条約機構（NATO）、カナダ、オストラリアなどの同盟国の軍事用であるた、一部の民間用として公開されていたものには、測位の精度を故意的に劣化させる操作（SA:Selective Availability）が発動させていたが、2000 年 5 月 2 日にそのＳＡは解除され、それ以前は 100m程度であった単独測位の制度は 10m程度まで大幅に改善された。また、モバイル通信の普及、GPS 受信機に小型化高性能とともに利用分野も大きく広がり、車、船舶、航空機等の移動局の位置測位だけではなく、地殻変動や地形の測量など、今後もその利用はますます増加すると考えられる。

現在では、GPS が無く陸、海、空に渡る交通輸送機関は安全で効率的な運行は危険であると言っても過言ではないほどになっている。さらには地震予知などのための地殻変動監視システムは GPS を全面的に頼っている。

2005 年 12 月 15 日はブッシュ大統領が新 PNT（測位、航法、タイミング）政策の承認を発表した。この新しい PNT 政策により、GPS とその関連サービスの将来的な利用、管理および改良に関する方針が明確になった。これは、ユーザーへの GPS の可用性を確実にし、GPS 業界の明るい未来を約束するものである。この政策が GPS をグローバルな情報技術標準として安定させるための道を開き続けるということである。10 年間で 2 度目の大統領声明として、PNT 政策は世界中の GPS ユーザからのダイナミックなニーズに応えるための安定したグローバルな政策環境の形成に関する米国の方針を示している。

1.2　研究の目的及び背景

ＳＡの解除以降、ＧＰＳ単独測位精度は10m程度になったが、ＤＧＰＳ（ディファレンシャル）ではその精度が2/3m、そして条件がよければ1m以下まで上げることができる。つまり、単独測位よりもＤＧＰＳのほうがより効率の良い測位方式なのである。現在、海上保安庁により全国で27の基準局（正確な位置が分かっている）が管理運営され、中波ビーコン電波及びFM放送の音声部分にDGPS補正データが多重されている。この補正データは広く利用され、日本全国でＤＧＰＳ測位を可能にしている。また、ＤＧＰＳをリアルタイムで行う技術を有している製品はもう市場には出ている。しかし、本研究室では学内に基準局を設け、補正値を計算し、後処理でＤＧＰＳ測位を行っている。今回、研究室のDGPS測位プログラムをリアルタイム化することを試み、本研究に取り組んだ。

本研究では、補正値を送る手段として無線ＬＡＮを使用した。データの通信プログラムをVISUAL C++で書き、測位計算プログラムはＣで書いてある研究室所有のプログラムを使った。本研究を通して次のことを達成するのが目的としている。

１）無線ＬＡＮの使用可能な範囲を調査すること。無線ＬＡＮの届く範囲、無線ＬＡＮを有効に使うために必要な周りの状況などを調べること。

２）Ｃ及びVISUAL C++でシリアル通信プログラム、DGPS測位プログラムを作成すること。

３）学内の基準局の補正値で、リアルタイムのＤＧＰＳ測位を行い、測位精度を評価すること。

1.3　GPS の概要

GPS システムは以下より構成されています。

1）宇宙部分（6 つの異なる軌道に 24 の GPS 衛星）

2）制御部分（5 つのモニター局、1 つの主制御局、３つのアップロード局）

3）ユーザ部分（GPS 受信機）

GPS 衛星は、図 1-1 の衛星軌道概念図のように、高度約 20,000km、傾斜角 55°軌道間隔 60°の 6 軌道を回る周回衛星であり、各軌道に 4 個ずつの計 24 個で運用するとされている。現在では、GPS 衛星は 28 個の衛星が運用しており、周回の周期は 12 時間弱（11 時間 58 分 02 秒）で、一年で同じ配置に戻るように推移しているようである。実際には軌道が随時変更されており決まってはいないが、衛星の軌道は 2m の範囲で管理されており、2 時間間隔で更新される高い精度を持った軌道情報を対応した衛星が、30 秒周期で放送している。

また、GPS では GPS タイムというものがあり、この時刻のもとに運用されている。各衛星には、セシウム原子時計または、ルビジウム時計が搭載されており、この原子時計は地上の官制システムで管理されており、この補正情報を軌道情報とともに衛星が放送している。GPS タイムは一週間で一回りするようになっている。また世界標準時である UTC が知られているが、UTC と GPS タイムの違いも管理されている。表 1-1 に GPS 衛星の主要緒元を記す。

表 1-1：GPS 衛星の主要諸元

衛星個数	4 個×６軌道面（4 個補助）
軌道半径	26,561km
衛星設計寿命	7.5 年
周回周期	12 恒星時間[約 11 時間 58 分 02 秒]
軌道傾斜角	55°
搬送波 周波数	L1=1,575.42MHz（10.23MHz×154） L2=1,227.6MHz（10.23MHz×120）
測距信号	C/A code:L1 で送信、民間にも開放 P（Y） code:L1,L2 で送信、非公開
送信電力	L1：C/A　Code：約 26W、P Code：約 13W L2：P　code：約４W
測位精度	水平方向約 10m
時刻精度	C/A code：約 20ns

各衛星は２つのＬ帯周波数即ちL1帯はC/A コード(Coarse Acquisition Code)とＰコード（Precise　Code)、L2帯はＰコードを２相位相変調して搬送する。C/Aコードはコードパターンが公表されているため利用できるが、Ｐコードはその周期が間引きなどで一週間に合わされていて、民間利用者向けではない。現在では、軍事用のＰコードが公表されたため、更に秘密性の高いＹコード（Ｐコードに含まれる情報を暗号化したもの）が開発され米軍関係者のみが利用可能となっている。

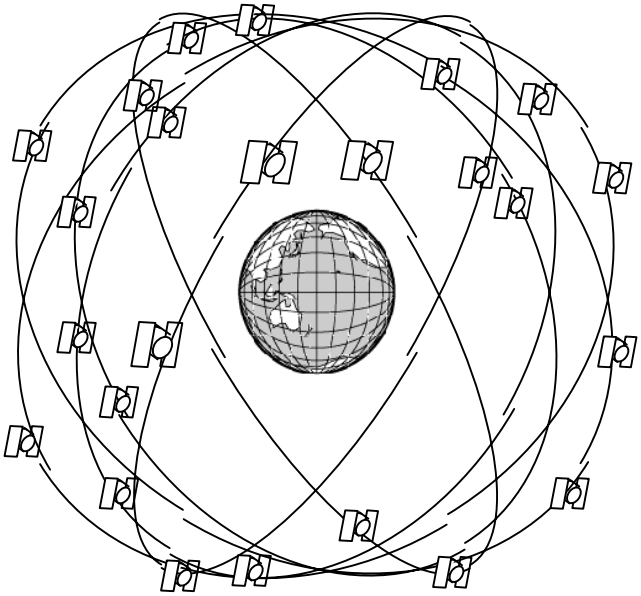


図 1-1： 衛星軌道概念図

全衛星が同じ周波数で送信しているにも関わらず混信しないのは、両波ともにゴールド符号と呼ばれる擬似ランダム雑音（PRN：Pseudo　Random　Noise）でBPSK変調（Bi－Phase　Shift　Key、２相位相変調）されており、衛星ごとに異なる符号パターンを割り当てることで、どの衛星かを識別できるからである。

1.4　DGPSの概要

DGPSシステムは図１-２で示しているように衛星、基準局、移動局及び両局の間にデータ送受信用のデータリンクの４つの部分から成り立っている。単独測位の精度を決める要因は衛星信号の精度、衛星の配置、電波伝搬路、C/Aコード位相の測定精度、測位演算の精度などである。通常は最初の３項目の影響が大きく、以下の項目はあまり影響しない。大気や電離層による誤差は衛星の仰角が極端に低くなれば同じ地域では概

略同じである。そして、衛星信号の誤差も同じである。このような性質を利用して精度を高めるのが DGPS(Differential GPS)測位である。

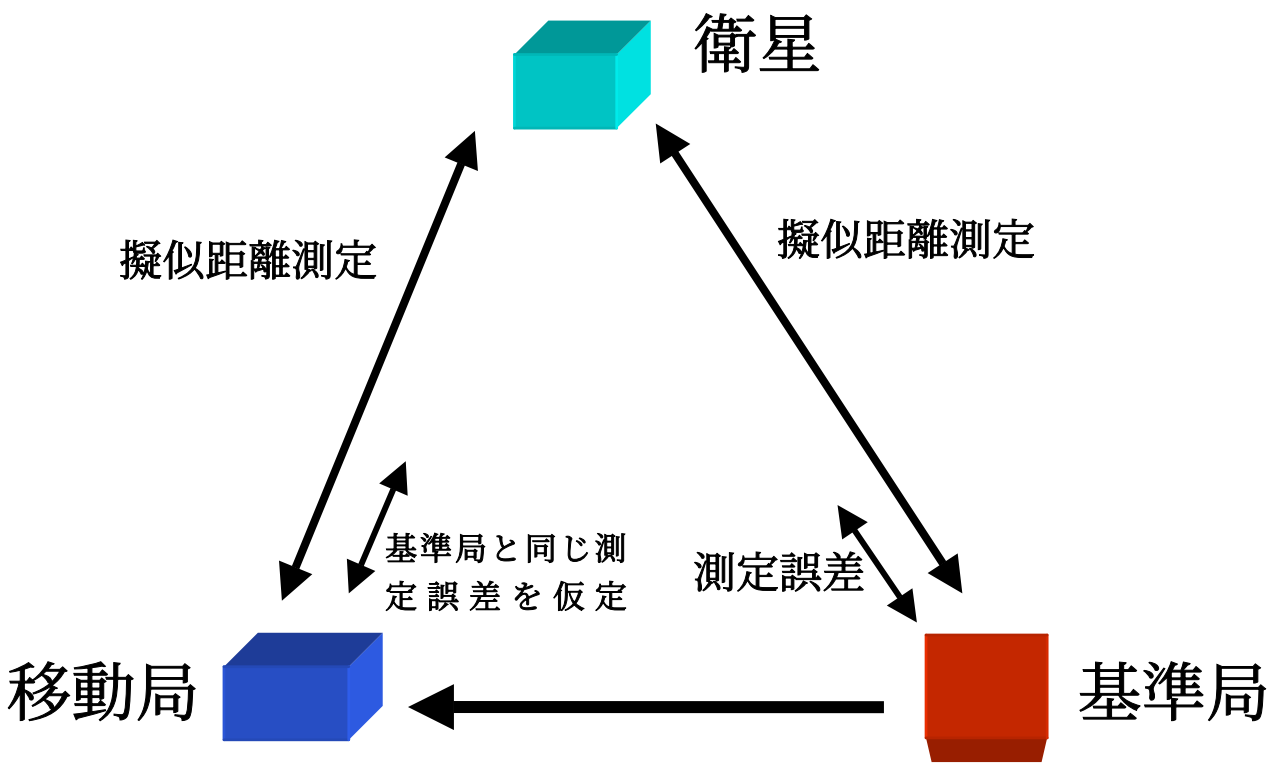


図 1-2：DGPS の仕組み

2000 年の SA 解除後、単独測位の精度は 10m 程度であるが、DGPS によって精度を１ｍまで上げることが可能になった。DGPS によって、基準局及び移動局に共通な誤差だけ補正ができ、マルチパスまたは受信機内部のノイズによる誤差の補正はできない。DGPS の精度は基準局と移動局間の距離に大きく影響される。

第 2 章　　測位原理及び測位計算

2.1　単独測位原理

単独測位は、GPS の標準的な利用方法であり、全世界で宇宙、航空、海上、地上（地中と水中は無理であるが）のどこでも、随時、１秒またはそれ以下の時間に現在地を知ることができる技術である。

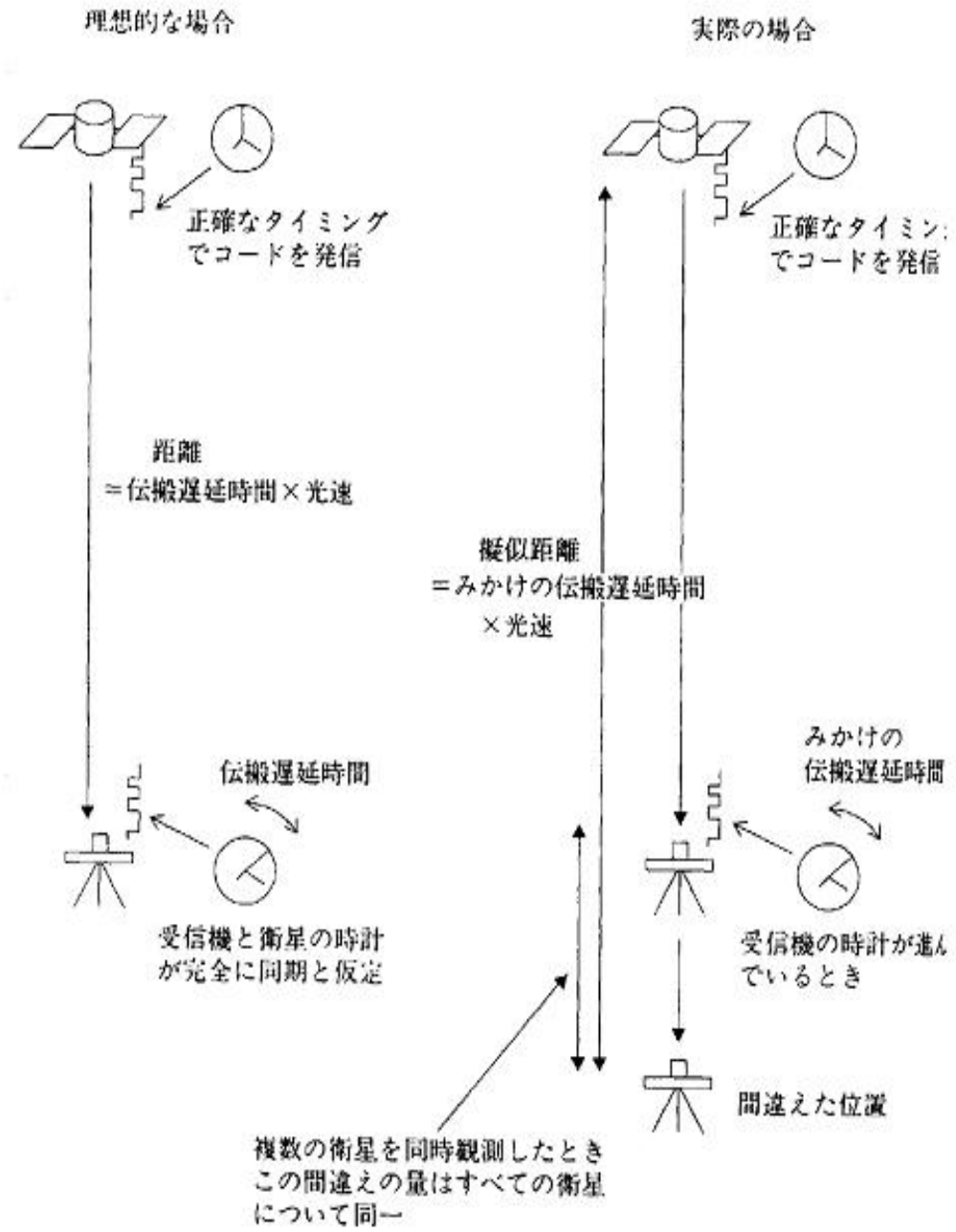


図 2-1：GPS 信号（C/A コード）による距離測定

GPS による測位は、利用者と衛星の距離を測定することによって行う。宇宙空間で位置の分かっている人工衛星三個のそれぞれの位置が分かっていれば測位できる。図 2-1 のように距離は衛星上の原子時計によって正確なタイミングで発信される測位用の符号C/AコードまたはPコードが受信機に到達するまでにどれだけ時間がかかったかを測定して求める。もしも、受信機の時計が衛星の時計と完全に同期していれば、その所要時間(伝搬遅延時間)が分かる。受信機と衛星の時計を完全に同期させることは、ほとんど不可能である。そこでもう一つの衛星が必要となる。受信機の時計によってコードの到達時刻を測定して、形式的に伝搬遅延時間を計算して距離を求める。これによって得られた距離は時間誤差と真の距離との和で、この距離のことを擬似距離という。

測位のために使用するそれぞれの擬似距離にはすべて同一の受信機の時計誤差が含まれている。図 2-2 で示したように、地球の中心を原点とした３次元座標（ｘ、ｙ、ｚ）と受信機の時計誤差Δｔの４個の未知数を解くので、４個以上の衛星を同時か、同時と見なせるだけの短時間に測定することが大切である。

また、GPS で求められる高さは、WGS-84 測地系であるのに対して、測位演算で出てくるものは３次元直交座標である。

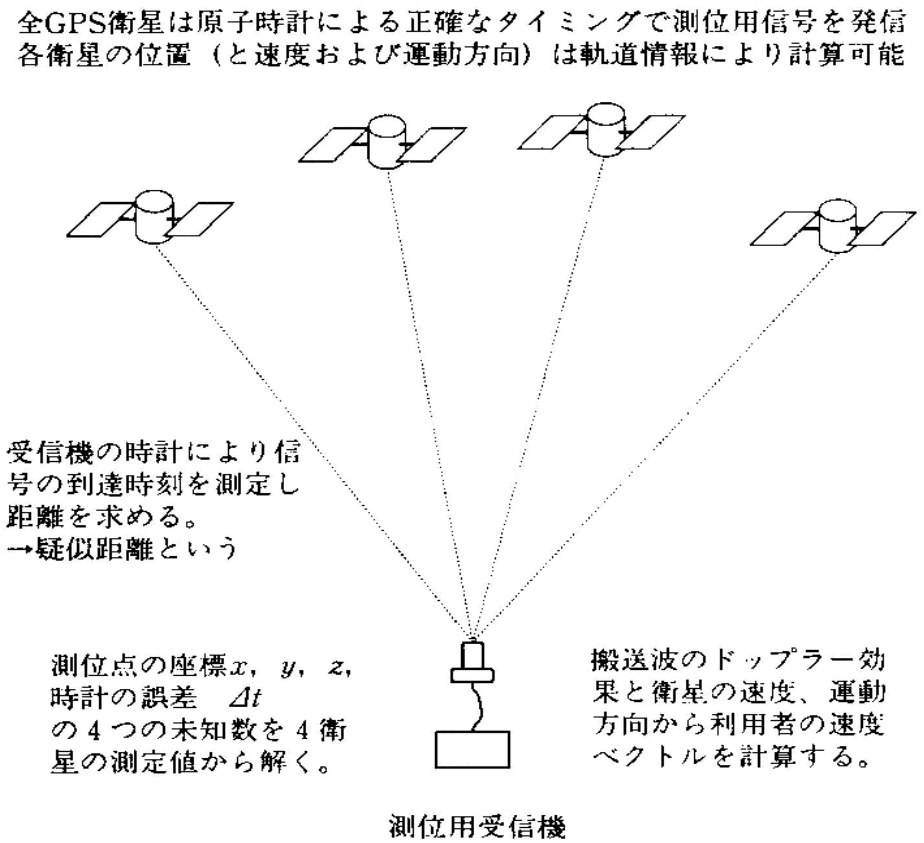


図 2-2：単独測位

2.2　単独測位演算

単独測位の測位計算は受信機で測定された各衛星との擬似距離をもとにして行う。４個の GPS 衛星の位置は、軌道情報から計算される。ここで用いる３次元直行座標系は、地球の中心を原点、地球自転軸に沿って北方向を正方向とする Z 軸、グリニジ子午面と赤道面との交点方向に X 軸、これら 2 軸と右手系をなすように Y 軸を選ぶものとする。

この座標系で、アンテナ位置の座標を、(x_0, y_0, z_0)とする。i 番の衛星の位置を (x_i, y_i, z_i) とする。測点と i 番目の衛星との距離は、ピタゴラスの定理によって、

$$\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} = c \times (\tau_i + \partial\tau) \qquad (2.1)$$

ここで τ_i は測定で得られた i 番目の衛星の電波伝搬時間であり、$\partial\tau$ は受信機の時計のずれ（正しい時刻より送れている時を正とする）で C は光の速度であり、
この式（２．１）の関係が観測している全ての衛星について成り立つ。
また観測点・衛星からの真の距離を r_{0i} とおく。

幾何学的には、(x_i, y_i, z_i) を中心とした半径が $c \times (\tau_i + \partial\tau)$ の３球面の交点として測点が求められる。

実際上の（２．１）式を解くことは二乗や平方根があり、うまく解くことができない。そこで未知数を近似値と補正量の和であらわし、最小二乗法を使用して解く方法がとられる。

i 番衛星とアンテナ位置との距離（擬似距離）r_i は、

$$r_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} \qquad (2.2)$$

と表すことが出来る。また、未知数をその近似値と補正量の和で表すと、

$$\left.\begin{aligned} x_n &= x_{n-1} + \Delta x \\ y_n &= y_{n-1} + \Delta y \\ z_n &= z_{n-1} + \Delta z \end{aligned}\right\} \qquad (2.3)$$

と表すことができ、これを式（２．１）に代入すると、

$$r_i = r_{n,i} + \frac{\partial r_i}{\partial x_i} \times \Delta x + \frac{\partial r_i}{\partial y_i} \times \Delta y + \frac{\partial r_i}{\partial z_i} \times \Delta z + s$$

（２．４）

ここで出てくる $\partial r/\partial x$ 、$\partial r/\partial y$、$\partial r/\partial z$ は次の式のようになり、それぞれ衛星方向のベクトルのｘ、ｙ、ｚ方向への射影（方向余弦）となっていることが分かる。

$$\left.\begin{aligned}
\partial r/\partial x &= \frac{x-x_i}{\sqrt{(x-x_i)^2+(y-y_i)^2+(z-z_i)^2}} \\
\partial r/\partial y &= \frac{y-y_i}{\sqrt{(x-x_i)^2+(y-y_i)^2+(z-z_i)^2}} \\
\partial r/\partial z &= \frac{z-z_i}{\sqrt{(x-x_i)^2+(y-y_i)^2+(z-z_i)^2}}
\end{aligned}\right\} \qquad （２．５）$$

ここで、 $\alpha=\partial r/\partial x,\quad \beta=\partial r/\partial y,\quad \gamma=\partial r/\partial z$ とおき

$$\Delta r = r_n - r_{n-1}$$

とおくと、ｉ番目の衛星からの擬似距離の補正量Δｒは

$$\Delta r_i = \alpha\,\Delta x + \beta\,\Delta y + \gamma\,\Delta z + s \qquad （２．６）$$

となり、4 衛星分の 4 個の方程式ができる。

$$\begin{pmatrix}\Delta r_1\\ \Delta r_2\\ \Delta r_3\\ \Delta r_4\end{pmatrix}=\begin{pmatrix}\alpha_1 & \beta_1 & \gamma_1 & 1\\ \alpha_2 & \beta_2 & \gamma_2 & 1\\ \alpha_3 & \beta_3 & \gamma_3 & 1\\ \alpha_4 & \beta_4 & \gamma_4 & 1\end{pmatrix}\begin{pmatrix}\Delta x\\ \Delta y\\ \Delta z\\ s\end{pmatrix} \cdots\cdots\cdots\cdots （２．７）$$

具体的な計算の求め方は、最初に x_o,y_o,z_o を式（２．２）に代入し近似距離 r_i を求める。実測された 4 衛星に対する擬似距離の差を Δr_i として、式（２．７）に代入してΔｘ、Δｙ、Δｚ、とｓを求める。これらの値から初期値である x_o,y_o,z_o にそれぞれ加え新しい初期値 x_o,y_o,z_o を作成する。作成した初期値を式(2-2)に代入し新しい近似距離 r_i を求め、新しい擬似距離との差 Δr_i を用いて新しいΔｘ、Δｙ、Δｚ、とｓを求めるという作業を繰り返すことにより、Δｘ、Δｙ、Δｚ、は 0 に収束し、これらが充分小さな値になった時点で計算を打ち切る。この時の初期値 x_o,y_o,z_o を求められたアンテナ位置とする。またＳは一定値に収束する。

また、実際の計算では、衛星が４個ではなく 4 個以上見えているので、（２．７）の式の両辺に転地行列をかければよい。仮にｎ個の衛星が見えているとすると、

$$A = \begin{pmatrix} \alpha_1 & \beta_1 & \gamma_1 & 1 \\ \alpha_2 & \beta_2 & \gamma_2 & 1 \\ \alpha_3 & \beta_3 & \gamma_3 & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \alpha_n & \beta_n & \gamma_n & 1 \end{pmatrix} \qquad \delta R = \begin{pmatrix} \Delta r_1 \\ \Delta r_2 \\ \Delta r_3 \\ \cdot \\ \cdot \\ \Delta r_n \end{pmatrix} \qquad \delta X = \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \\ s \end{pmatrix}$$

$$\delta R = A \cdot \delta X$$

となる。両辺に A の転置行列を掛けると、

$$A^T \cdot \delta R = A^T \cdot A \cdot \delta X \qquad （２．８）$$

となる。次に両辺に A と A の転置行列をかけた行列の逆行列をかけると、

$$\delta X = \left(A^T \cdot A\right)^{-1} \cdot A^T \cdot \delta R \qquad （２．９）$$

となり補正量Δx、Δy、Δz、s を求めることができる。

2.3　DGPS 測位原理

DGPS は予め位置が正確に分かっている固定基準局(refernce staition)で測位誤差を求め、それによってユーザー局（移動局）での測位結果を補正して精度を改善するものである。その精度は２メートル程度で、条件がよければ１メートル以内の測位精度となる。ただし、DGPS 測位では基準局とユーザー局において大気や電離層等の遅延量を同じと仮定しているので両地点間の最大距離は数 100ｋｍから数 1000ｋｍまでであり、それ以上になると精度が低下するので注意が必要となる。また、基準局からの補正データを伝送する必要があるためその伝送方法と距離が測位精度に影響することとなる。以下に DGPS 測位で利用されている 2 種類の補正方式について述べる。

2.3.1　測位位置補正方式

この方式は、正確に分かっている基準局の位置と、同地点で単独測位により出力された位置を比較することによって、位置の補正値(緯度、経度、高さもしくは三次元直交座標系の xyz 座標)を算出し、ユーザー局の測位結果にこの補正値を適用するものである。以下の数式によって測位位置補正方式を表す。

$$\mathrm{X}_u = \mathrm{X}'_u - \left(\mathrm{X}'_r - \mathrm{X}_r\right) \tag{2.10}$$

$\mathrm{X}_r, \mathrm{X}'_r$ はそれぞれ基準局の正確な位置座標と単独結果を表し、$\mathrm{X}_u, \mathrm{X}'_u$ は移動局での DGPS 測位結果と単独測位結果を表している。移動局側では通常の GPS 受信機が出力する単独測位結果を使用でき、計算処理も単純である。しかし、基準局側と移動局側で同じ組み合わせの GPS 衛星を使用して測位を行わなければならないという制限がある。でも次に述べる擬似距離補正方式と比較して、基準局からユーザー局へ伝送するデータ量は格段に少なくてすむ。

2.3.2　擬似距離補正方式

この方式は、基準局で受信している衛星までの真の距離と、ユーザーが実際に測定した擬似距離を比較することによって、衛星ごとの擬似距離補正値、及びその変化率を算出してユーザー局に伝送する。ユーザー局では受信機で測定された擬似距離に、この補正値を適用して測位するものである。

前述した測位位置補正方式と異なり、基準局で擬似距離補正データが得られた衛星であれば、ユーザー局ではどんな衛星の組み合わせを測位に使用しても良いという柔軟性がある。この方式は世界標準として広く利用されている。本研究でもこの方式を使いましたが、測位用の PC を一台しか使ってないので、基準局の擬似距離情報を全て無線 LAN で送信し、移動局にある測位計算用の PC で計算し、補正値を計算し、移動局の擬似距離にこの補正値を使い DGPS 測位を行う形になっている。

第３章　　システムの構成

3.1　現在、研究室で使われているシステム

DGPS の概要で説明したように、DGPS 測位には基準局の単独測位によって計算される補正値を使う。現在、研究室では基準局及び移動局での GPS 生データをファイルに落して、両方のファイルを同じ PC に入れて読み込んでいる。基準局の GPS 時刻及び移動局の DGPS 時刻が一致したところで、基準局の単独測位を行い、補正値を計算し、これらの補正値を使い移動局の DGPS 測位を行っている。これは後処理ファイル方式でその流れ図を図１に示す。

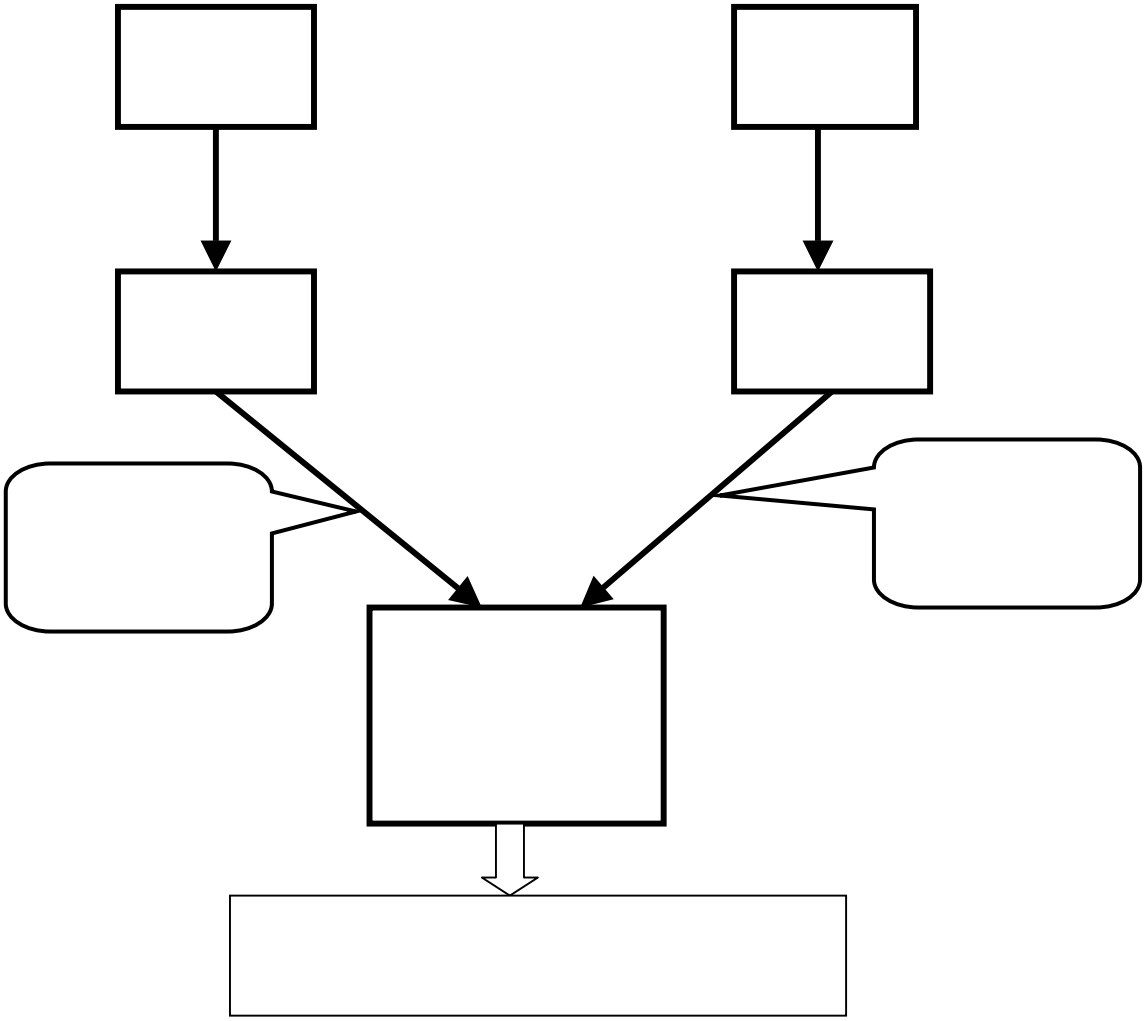


図 3-１：後処理による DGPS 測位システム

3.2　DGPS 測位システムのリアルタイム化

後処理方式の DGPS 測位では、両局の生データをファイル化しているが、リアルタイム化するためにはある一定時間のデータをファイル化する代わりに、一定時間毎（例えば１秒毎）のデータをリアルタイムで受け取ることが必要である。移動局の PC は移動局の受信機からデータをシリアル（RS232C）で受け取り、尚且つ基準局のデータも何らかの方法で受け取らなくてはならない。そこで、今回基準局の生データを無線 LAN で送信し、無線 LAN を使える PC で受け取る方法を考えた。つまり、PC 上のシリアル RS232C ポート（通常 COM ポート）から移動局の生データを取ると同時に無線 LAN で受信した基準局の生データを仮想 COM ポートから取ることで、DGPS 測位をリアルタイムで行うということである。COM ポート（通常及び仮想）について次のページに説明する。リアルタイムで DGPS 測位システムの流れを図２に示す。

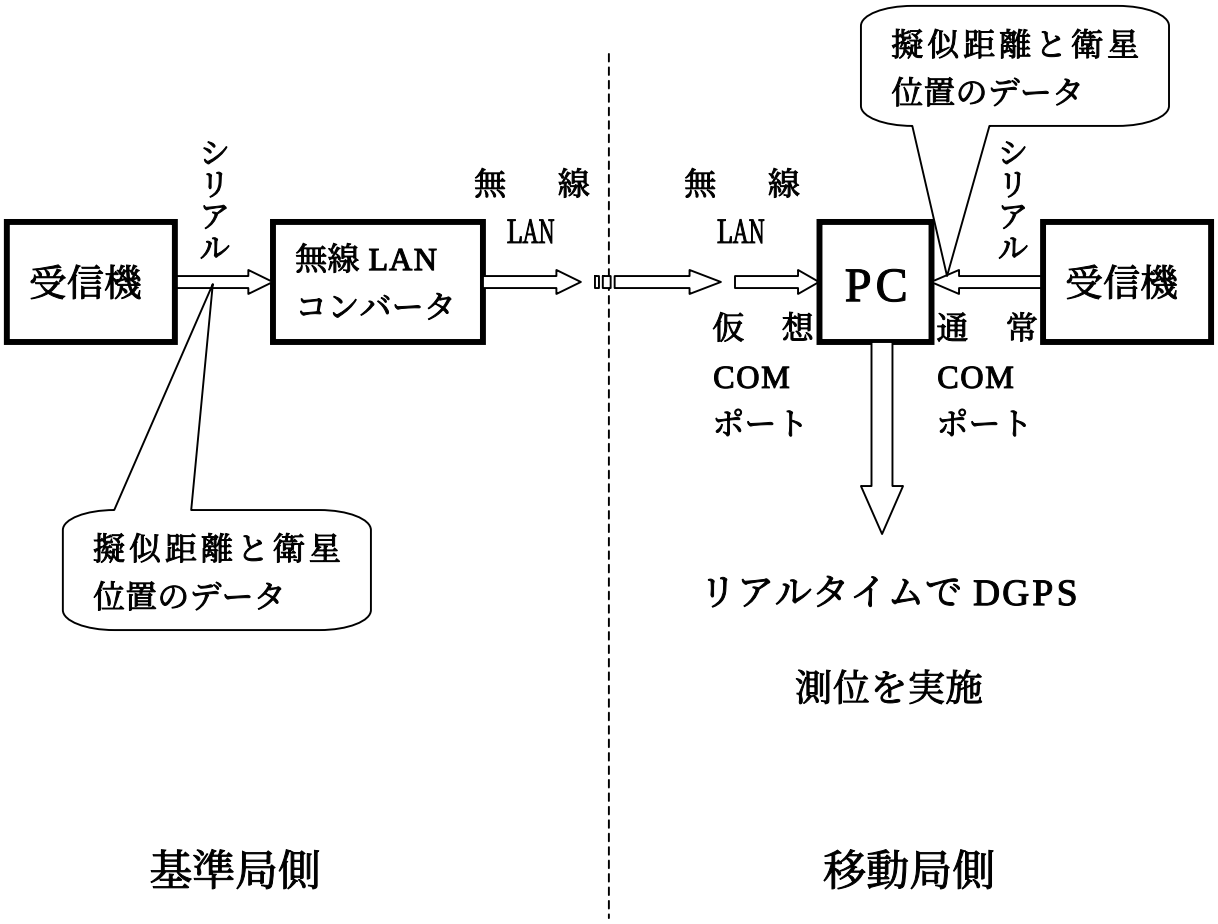


図 3-2：リアルタイムでの DGPS 測位システム

3.3　通常及び仮想 COM ポート

3.3.1　COM　ポート

COM ポートは COMMUNICATION　ポートの略で PC/AT 互換機（IBM 社が 1984 年に発売したパソコンと互換性のあるパソコンの総称）と外部通信機器（プリンタ、モデム、スキャナなど）との間でデータの送受信のためのパソコン側のコネクタまたはインターフェースのことである。

3.3.2　通常 COM ポート

通常 COM ポートとは、外部通信機器との送受信を行うためにパソコン側に元々ついている COM ポートのことを意味している。データ転送の方式によって２種類の COM ポートがある。

１）シリアル COM ポート（SERIAL COMMUNICATION PORT）

シリアル COM ポートはパソコン本体と周辺機器間のシリアルデータ転送用のポートである。シリアル転送というのは、１本の信号線を使って１ビットずつデータを転送する方式であり、シリアル転送方式では一方転送及び双方向転送がある。一方転送の場合は２つの信号線を使用する。一つが送信用で、もう一つはグラウンドとなる。しかし双方向転送の場合は信号線が３つで送信と受信のため別々の２つの線があって、もう一つはグラウンドになる。シリアル転送方式はパラレル転送方式に比べ、転送速度は遅いが、最大伝送距離は長い。パソコン上のシリアルポートはモデム、キーボード、マウスなどの接続に使われる。シリアル COM ポートの例を幾つか挙げると RS－232C,RS－422（Macintosh のシリアルポート）、USB、IEEE－1394 などがあってその中で最も普級しているのは **RS232C** である。

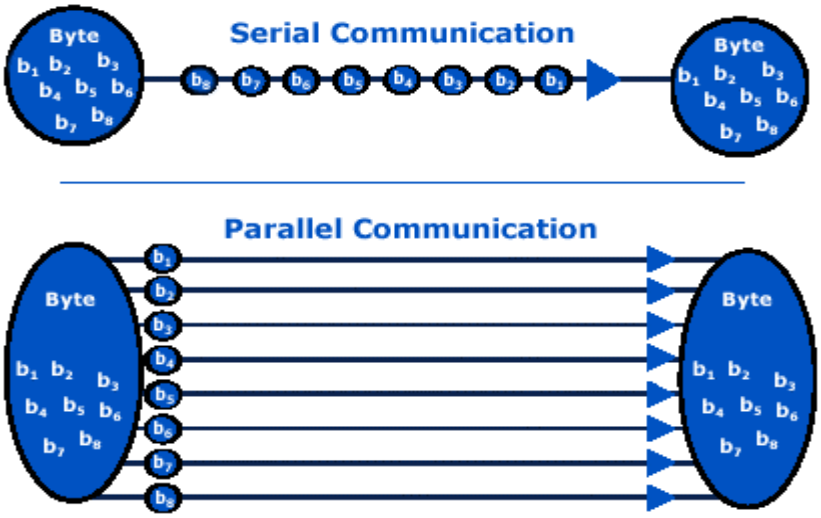


図 3-3：シリアルとパラレルデータ転送方式

RS－232　は RECOMMENDED　STANDARD の略で、アメリカの EIA（ELECTRONIC INDUSTRIES　ALLIANCE、米国電子工業会）によって標準化された、シリアル通信規格の

一つでＣはこれらの最新バーションの意味である。これはほとんどのパソコンに標準で搭載されている。ケーブルの最大長は約 15m で、最高通信速度は 115.2kbps。コネクタには D-sub25 ピンのものもあるが、最近のほとんどの PC には D-Sub9 ピンのものが使われている。本研究では通常 COM ポートとしてこの RS－232C　というシリアル COM ポートを使用した。RS－232C　用の D－Sub９ピンの役割を表１に示す。

表 3-1：RS232C　DB９ピンの配置と役割

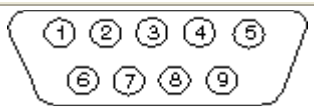
	9 Pin Connector on a DTE device (PC connection)
Male RS232 DB9	
Pin Number	**Direction of signal:**
1	Carrier Detect (CD) (from DCE) Incoming signal from a modem
2	Received Data (RD) Incoming Data from a DCE
3	Transmitted Data (TD) Outgoing Data to a DCE
4	Data Terminal Ready (DTR) Outgoing handshaking signal
5	Signal Ground Common reference voltage
6	Data Set Ready (DSR) Incoming handshaking signal
7	Request To Send (RTS) Outgoing flow control signal
8	Clear To Send (CTS) Incoming flow control signal
9	Ring Indicator (RI) (from DCE) Incoming signal from a modem

表１では DTE　と DCE はそれぞれパソコンと外部通信機器のことである。ここで、RTS/CTS ピンはハードウエアーフローコントーロル用で、ソフトウエアフローコントーロル（Xon/Xoff）が選択されているときは動作しない。

２）パラレル COM ポート（PARALLEL COMMUNICATION PORT）

パラレル転送はシリアル転送と違って、8 本の信号線を使って同時に８ビットすなわち１バイトのデータを転送する方式である。転送速度は高速だが、最大伝送距離は短くなる。パソコン用のパラレル転送方式としては、プリンタなどとの接続に使われるセントロニクス仕様や IEEE 1284 などが最も普及している。記憶装置の接続などに使われる SCSI（Small Computer System Interface）や IDE（Integrated Drive Electronics）もパラレル転送方式の一種である。

3.3.3　仮想 COM ポート

ソフトウエアによって作成された、通常の COM ポートと同じ働きをするポートのことである。今回の研究では PC 上に仮想 COM ポートを作成するために使ったのはアルファプロジェクト社（ALPHA　PROJECT　CO.LTD.）から購入した TCP-VSP というアプリケーションソフトである。TCP-VSP は Transfer　Control　Protocol-Virtual　Serial Port の略で、ネットワークに接続された無線 LAN コンバータの TCP/IP ポートを PC 上の仮想 COM ポートとして使う。
この TCP-VSP ソフトの特徴を挙げると次のようになる。

１）ソケット通信プログラムの作成が不要

ソフトをインストルしてＣＯＭポートの設定を行うだけで当社の ezTCP シリズの無線ＬＡＮコンバータ（今回使ったのはＥＺＬ３００Ｗ）との通信を行えるので、新たにソケット通信の知識を学ぶ必要はない。

２）既存のシリアル通信設備をＬＡＮ環境に移行可能

ＰＣ上では通常のＣＯＭポートと同じ取り扱いになるので、ezＴＣＰとＴＣＰ－ＶＳＰを導入するだけで既存のシリアル通信設備をそのままＬＡＮ環境に移行できる。

３）最大 256 個の仮想ＣＯＭポートの作成/通信が可能

最大 256 個の仮想ＣＯＭポートを作成し、ＴＣＰ/ＩＰポートとの通信が行える。注意しなければならないのは、ＰＣ上で既に割り当てられている通常ＣＯＭポート（ポート番号）に仮想ＣＯＭポートを割り当てることはできない。
TCP/VSP のメインウィンドウは図３のようになる。

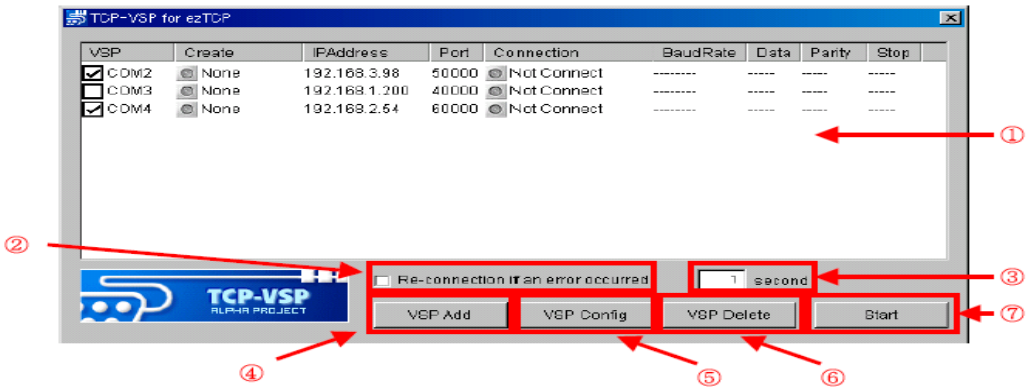


図 3-４：仮想ＣＯＭポートの実行画面

画面上の表示項目及びボタンの説明：

１）仮想ＣＯＭポートの状態表示

1. VSP:仮想ＣＯＭポートと使用/未使用の表示（チェックが付いていると使用できる）及びポート番号
2. Create：仮想ＣＯＭポートの動作状態。None、Create Error、Create　の三種類があってそれぞれポートが作成されてない、作成エラー、作成成功の意味を表す。
3. IPAddress：接続先すなわち無線ＬＡＮコンバータの IPAddress のこと。
4. Port：接続先のポート番号である。
5. Connection：TCP/IP の接続の状態を表し、これの表示メッセージーを表 2 に示す。
6. BaudRate：通信のボーレート。ＯＥＭ３とＯＥＭ４の受信機ではそれぞれ５７６００と１１５２００を使用した。
7. Data：データビットの長さで８に設定した。
8. Parity：パリティ表示である。０を使用した。
9. Stop：ストップビットで１に設定した。

２）“Reconnected If An Error Occurred”チェックボックス

これをＯＮにすると TCP/IP 接続エラー時に再接続を行う。

３）時間設定のエディットボックス

再接続を行う設定のみ有効で、再接続時にもエラーが発生したときに次の再接続までの間隔秒単位で設定する。

４）“VSP Add” ボタン

仮想ＣＯＭポートを追加するための設定ウィンドウを開く。

５）“VSP Config”ボタン

仮想ＣＯＭポートの編集を行うためのウィンドウである。

６）“VSP Delet”ボタン

選択されている仮想ＣＯＭポートを削除する。

７）“Start”ボタン

仮想ＣＯＭポートの処理を開始する。

８）“Stop”ボタン

仮想ＣＯＭポートの処理を終了する。

表 3-2：CONNECTION 項目のメッセージ一覧

表示メッセージ	意味の説明
Not Connection	接続してない状態
Create Error	TCP/IP 通信で使用する Socket 作成エラー
Connect Error	TCP/IP 通信の接続エラー
Not Support Device	接続先が ezTCP シリズではない
Connecting	接続の処理中
Connected	接続成功及びデータ通信中
Reconnecting	再接続の処理中
Reconnected	再接続成功及びデータ通信中

仮想ＣＯＭポートの追加及び編集

TCP/VSP のメインウィンドウを開き、VSP Add または VSP Config ボタンをクリックすると図 3-5 の画面が表示される。

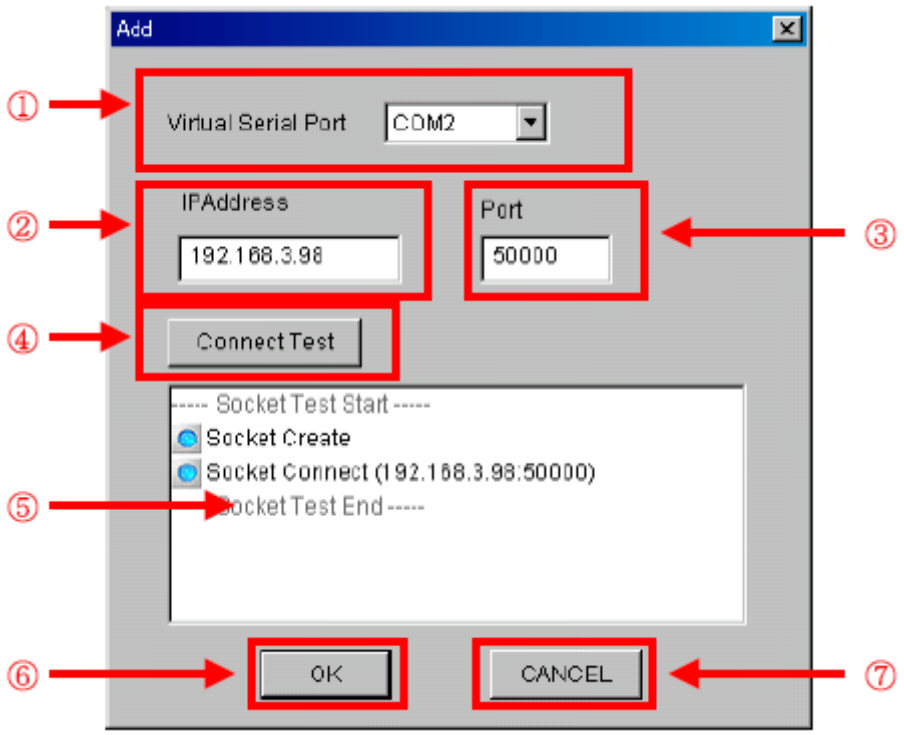


図 3-5：仮想ＣＯＭポートの追加/編集画面

①のリストボックスから通常ＣＯＭポートで割り当てられていない番号(仮想ＣＯＭポート番号となります）を選択し、②と③の部分でそれぞれ無線ＬＡＮコンバータ（ezTCP）の IPAddress とポート番号を入力し、“ Connect Test ”ボタンを押して接続テストを行います。テスト結果は⑤のエディットボックスに表示される。接続できない場

合は“Socket Connect(IPAddress.Port)”のところは赤くなる。このときは、使っているＰＣの無線ＬＡＮの接続及び無線ＬＡＮコンバータ（ezTCP）は正常に動いているかを確認してもう一度テストを行う。接続が成功したら、ＯＫボタンをクリックすると、作成完了の仮想ＣＯＭポートの動作中の画面（図５）が開く。

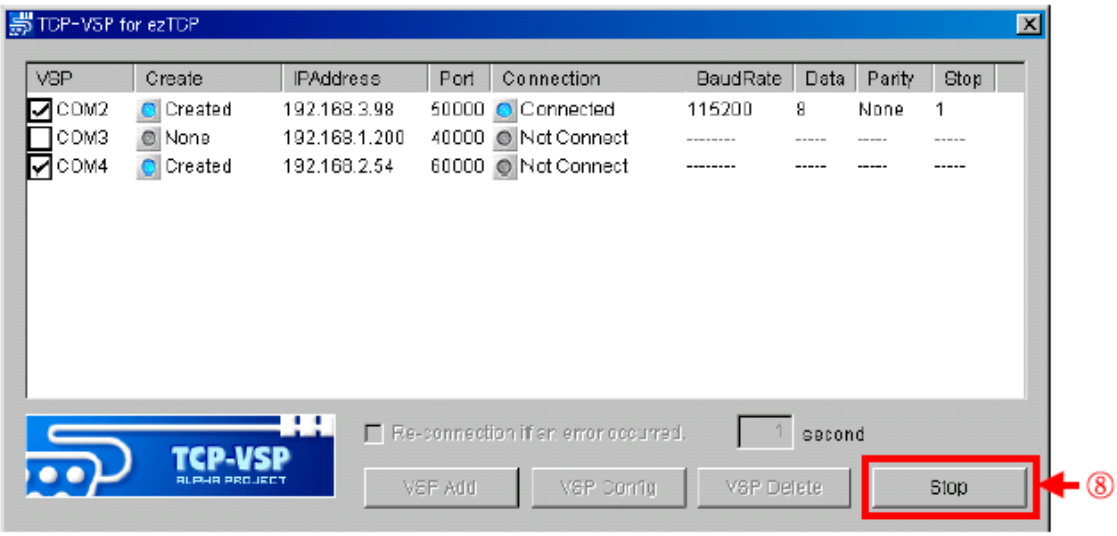


図 3-6：仮想ＣＯＭポートの動作中の画面

図 3-６を見ると仮想ＣＯＭポート２と４が作成されていて、両方のＣＯＭポートを使いたいので左のチェックボックスにチェックを入れて“Start”ボタンを押した後、仮想ＣＯＭポート２だけが接続されて４は接続できなかった事が分かる。今の状態では、仮想ＣＯＭポート２を無線ＬＡＮからデータを受信するためにプログラムの中では通常のＣＯＭポートと同様に扱うことができる。図５では、Start ボタンが Stop ボタンに変わっている。ポートの接続中 Stop ボタンを押すと仮想ＣＯＭポートが切断される。

第４章　通信及び測位プログラム

4.1　プログラム言語の概要

プログラムは主にVISUAL　C++(以降はVC++と書く)で作成しましたが、中にはC/C++で書いた測位用のプログラムも入っている。本研究を通してVC++について勉強したので、このプログラムに使ったVC++の基本的なところについて以下説明する。

VC++は実際にはMicrosoft　Developer　Studioという開発環境のことを示し、MSDN（Microsoft　Developer　Network）は、VC++を含むMicrosoftの全てのプログラム製品のオンライン　ヘルプ　システムである。VC++の最も注目すべき技術の一つは、Wizardである。各Wizardは、特定の種類のプログラムのためにプロジェクト（プログラムの全ての部分例えばソースファイル、ヘッダファイルなどの集合）の初期設定を行う。Wizard は、新規プロジェクトの作成時にプログラムのカスタマイズ可能な枠組を提供してくれるので、新規プロジェクトの作成はゼロから始める必要はない。VC++の代表的なWizardであるAppWizardは、Developer　Studio環境のダイナミック　リンクライブラリとして実行される。AppWizardの対話形式のインターフェイス（幾つかのステップ）に答えて新規プログラムで実現したい機能を設定すると、その機能を実現するために必要なコードを持つソースファイルが自動的に生成される。WINDOWSシステム全体の関数群のことをWin32　API（Application　Program　Interface）と言うが、その大部分を含めているのがMFC(Microsoft　Foundation　Class)ライブラリである。Win32の意味はWINDOWSプラットフォーム用の32ビットプログラミングのことである。16ビットプログラミングには、MFC1.5*x*を使うほうが良い。MFC3.0以降のフレームワークでは、Microsoft　Windows95　や　Microsoft Windows NT 3.51 以降などのWin32ビットプログラミングがサポートされている。MFC　AppWizardというプロジェクトでは次の3種類のApplicationがある。

１．SDI(Single　Document　Interface)：このアプリケーションでは一度に 1つだけ、ドキュメント フレーム ウィンドウを開くことができる。

２．MDI（Multiple　Document　Interface)：このアプリケーションでは、同じインスタンスで複数のドキュメント フレーム ウィンドウを開くことができる。MDIアプリケーションのウィンドウには、複数の MDI 子ウィンドウを開ける。MDI 子ウィンドウは、それ自身がフレーム ウィンドウであり、別々のドキュメントを保持する。

SDIとMDIの両方はAppWizardのドキュメント/ビューアーキテクチャ機能を有している。これは、プログラムのデータはドキュメントオブジェクトにより管理され、ビューオブジェクトを介してユーザーに表示される機能である。

３．ダイアログベースインターフェイス：このアプリケーションでは、ユーザーとのインターフェイスはドキュメント/ビューアーキテクチャではなく一つのダイアログボックスによって提供される。VC++では AppWizard を使わずにダイアログベースアプリケーションを作成することもできる。

今回のプログラムは MFC　AppWizard のダイアログベースインターフェイスによって作成したものである。VC++では、プロジェクトを作成する時 StdAfx.h のヘッダファイル及び StdAfx.cpp のソースファイルが自動的に作成される。StdAfx.h は標準のシステム インクルード ファイル、または参照回数が多く、かつあまり変更されないプロジェクト専用のインクルード ファイルを記述しているので、後からプロジェクトに追加する全てのソースファイルにこれをインクルードしなければならない。

クラス：

クラスは VC++の基本的な概念である。これは変数と関数が同居している集合体であって、これらの変数と関数のことをそれぞれ member 変数と member 関数と呼ばれる。クラスには private, protected と public の３つの部分があり、public 部に書かれることだけが外から使える。MFC はこのようなたくさんのクラスの集合であり、その一部のクラスは Win32 API の大部分をカプセル化している。ドキュメント、ビュー、アプリケーション自体などのアプリケーションの概念をカプセル化するクラスもあり、OLE の機能や ODBC や DAO によるデータへのアクセス機能をカプセル化するクラスもある。たとえば、Win32 のウィンドウの概念は、MFC の CWnd と言うクラスでカプセル化されている。つまり、CWnd という C++ クラスは、Windows のウィンドウを表す HWND ハンドルをカプセル化 ("ラップ") している。これと同様に、Cdialog と言うクラスは Win32 のダイアログ ボックスをカプセル化している。

カプセル化とは、たとえば C++ の CWnd クラスの中で HWND 型のメンバ変数を定義し、パラメータとして HWND を取る Win32 関数の呼び出しを CWnd クラスのメンバ関数にカプセルのように封じ込めるという意味である。クラスのメンバ関数は、通常、カプセル化した Win32 関数と同じ名前が付けられている。

ダイアログベースアプリケーションではプログラム 1 本に付きクラスが２つ作られる。今回は RS232 と言う名前のプロジェクトを作成したので、MFC の AppWizard 機能によって CRS232App（プログラム全体を統括する）と CRS232Dlg（ダイアログを表示する）と言う名前のアプリケーションクラスとダイアログクラスが生成された。これらのクラスは元々MFC にあるそれぞれ CWinApp と CDialog の親クラスから派生されるものである。

CWinApp クラス：これは Windows のアプリケーション オブジェクトを派生させるための基本クラスである。AppWizard は、以下のアイテムから成るインプリメンテーション ファイルを生成する。

1.アプリケーション クラス「CRS232App」のためのメッセージ マップ
2.空のクラス コンストラクタ「CRS232App()」
3.クラスの唯一のオブジェクトを宣言する変数「theApp」

4.メンバ関数 InitInstance ( ) の標準的なインプリメンテーション

アプリケーション クラスは、プロジェクトのヘッダーファイル（RS232.h）とメイン ソース ファイル（RS232.cpp）に置かれる。これらのクラスのコードを表示するには、プロジェクト ワークスペース ウィンドウの [ClassView] を使うのが最も簡単である。多くの場合、標準のインプリメンテーションおよびメッセージ マップをそのまま使用できるが、必要に応じて変更することもできる。標準インプリメンテーションの中でオーバーライドが必要になるのは、メンバ関数 InitInstance（）だけで、通常は、InitInstance（）のスケルトン インプリメンテーションにコードを追加する。

CDialog クラス：CDialog クラスは、画面上にダイアログ ボックスを表示するための基本クラスである。ダイアログ ボックスはウィンドウの特別な種類ですので、CDialog クラスは Windows の CWnd クラスから派生される。ファイルのオープンや保存、印刷、フォントやカラーの選択、検索-置換の起動、いろいろな OLE 関連の操作などを行うには、CDialog からダイアログ クラスを派生させるか、標準ダイアログ ボックスのためのコモン ダイアログ クラスの内の１つを使用することができる。このプログラムでは、AppWizard によって CDialog クラスから CRS232Dlg クラスが派生され、ヘッダファイル（RS232Dlg.h）と次の 6 つのメンバ関数を持つインプリメンテーションファイル（RS232Dlg.cpp）が生成される。

6. 標準コンストラクタである CRS232Dlg（）
2.DDX と DDV の呼び出しようの関数 DoDataExchange（）
3.ダイアログボックスの初期化設定用の関数 OnInitDialog（）
4.システムとのパラメータをやり取り用の関数 OnSystemCommand（）
5.ダイアログボックスにアイコンなどの描画用の関数 OnPaint（）
6.最初化ウィンドウをドラッグしている間にカーソル表示用関数 OnQueryDragIcon（）

ダイアログボックス

これは本研究のプログラムで一番重要なものである。ダイアログ ボックスは、ステータス情報の表示や、ユーザーが入力した情報の取得に使用されるウィンドウで、ユーザーはダイアログ ボックスを使ってプログラムと情報をやり取りする。ダイアログ ボックスには、以下のコントロールを作成することができる。

1.Windows コモン コントロール。たとえば、エディット ボックス、プッシュボタン、リスト ボックス、コンボ ボックス、ツリー コントロール、リスト コントロール、プログレス インジケータなど。

2.ActiveX コントロール

3.オーナー描画コントロール。このコントロールは、ダイアログ ボックスに描画する。

ダイアログには次の 2 タイプがある。

1）モーダルダイアログ ボックス：これはダイアログ ボックスが開かれている間は、そのダイアログ ボックスしか操作できない。モーダル ダイアログ ボックスを作成するには、CDialog で宣言されている 2 つのパブリック コンストラクタのどちらかを呼び出し、ダイアログ オブジェクトのメンバ関数 DoModal を呼び出して、ダイアログ ボックスを表示する。ユーザーが [OK] または [キャンセル] をクリックすると、入力処理が終了する。モーダル ダイアログ ボックスでは、DoModal がダイアログ リソースをロードする。これのコードは次のようである。

```
CRS232Dlg dlg;

    m_pMainWnd=&dlg;

    int nResponse=dlg.DoModal();

    if(nResponse==IDOK){

        // TODO: ダイアログが <OK> で消された時のコード

    }

    elseif(nResponse==IDCANCEL){

    // TODO: ダイアログが <ｷｬﾝｾﾙ> で消された時のコード

    }
```

2）モードレスダイアログ ボックス：この場合は、ダイアログ ボックスを開いた

まま、ほかのウィンドウで作業を行うことができる。モードレス ダイアログ ボックスを作成するには、そのダイアログ クラスに独自のパブリック コンストラクタを用意し、このパブリック コンストラクタを呼び出し、次にダイアログ オブジェクトのメンバ関数 Create を呼び出してダイアログ リソースをロードする。Create は、コンストラクタ呼び出し中でも後でも呼び出せる。ダイアログ リソースに WS_VISIBLE プロパティがあるダイアログは直ちに表示されるが、WS_VISIBLE プロパティがないダイアログはメンバ関数 ShowWindow を呼び出して表示する必要がある。コードは次のようである。

```
m_pModeless = NULL;

    if(!m_pModeless)

            m_pModeless = new CRS232nDlg;

    if(!::IsWindow(m_pModeless->GetSafeHwnd()))

            m_pModeless->Create(IDD_DIALOG1,this);

  m_pModeless->ShowWindow(SW_SHOW);
```

今回のプログラムでは、移動局用のダイアログは親ウィンドウでモーダルタイプであって、基準局用のダイアログボックスは子ウィンドウでモードレスダイアログである。作成したダイアログを図 4-1 に示す。

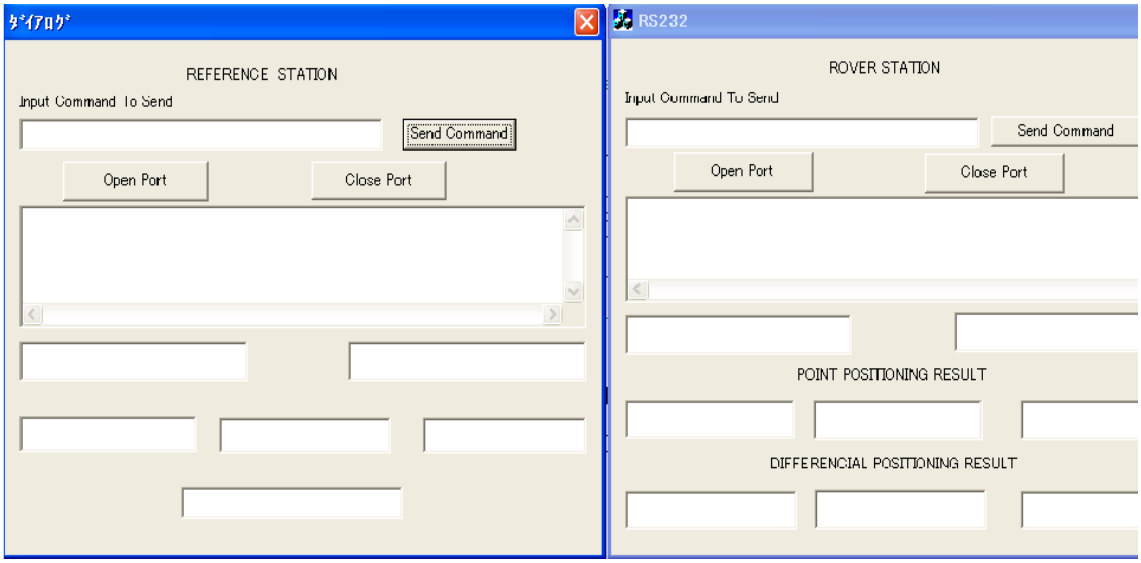


図 4-1：コントロールが付いている２つのダイアログ

4.2　通信プログラムについて

本研究では、測位用の PC の通常 COM ポート及び仮想 COM ポートによってそれぞれ移動局と基準局の受信機との通信を行うために PC の COM ポートを制御するプログラムが必要である。これは WIN32　API のシリアル通信関数を使って COM ポートの設定及び動作を制御している。COM ポートの動作及び制御用のヘッダ及びソースファイルはそれぞれ com.h 及び com.cpp である。

Com.h のヘッダファイルでは次の WIN32　API のシリアル通信関数を定義している。

1）int APIENTRY RS_Open(int p_no,long brate,int bsize,int sbits,int parity);
　　これは COM ポートの開き及びポートの色々な設定を行う
2）int APIENTRY RS_Close(int p_no);
　　これはポートを閉じる
3）long APIENTRY RS_GetInBufferCount(int p_no);
　　受信バッファに通信エラーが発生するときの対策を行い、バッファにあるデータの数（Queue のデータ数）を把握する
4）long APIENTRY RS_GetOutBufferCount(int p_no);
　　送信バッファに通信エラーが発生するときの対策を行、バッファにあるデータの数（Queue のデータ数）を把握する
5）int APIENTRY RS_Read(int p_no, LPSTR lpszStr, long len);
　　受信バッファからデータを読み込む
6）int APIENTRY RS_Write(int p_no, LPSTR lpszStr, long len);
　　送信バッファに送信したいデータを書き込む

次に通信を行うために一番需要である com.cpp のソースファイルに書いてある以上の関数の中身について簡単に説明する。

1）int APIENTRY RS_Open：この中に次に説明する関数によって COM ポートをオープンすることからポートの色々な設定を行っている。

A）**CreateFile 関数：**COM ポートへのハンドルとしてｈCom[ポート番号]を定義し、この関数を使ってポートを開いている。この関数は次のパラメータを持っている。

LpFileName：作成する (またはオープンする) オブジェクトの名前が入った NULL で終わる文字列へのポインタを指定します。ここでは、COM ポートを指定している char 型

ポインタ lpszComDeviceName[6]である。

DwDesiredAccess：オブジェクトへのアクセスの種類を指定し、次の値の組み合わせを指定する。

値	意味
0	デバイスの属性を問い合わせる。
GENERIC_READ	読み取りアクセスで、データの読み取りとファイルポインタの移動ができる。読み書きアクセスをするには、GENERIC_WRITE と組み合わせて指定する。
GENERIC_WRITE	書き込みアクセスで、データの書き込みとファイルポインタの移動ができる。読み書きアクセスをするには、GENERIC_READ と組み合わせて指定する。

DwShareMode：オブジェクトの共有方法を指定する。0 を指定したので、オブジェクトは共有されない。ハンドルをクローズしない限り、このオブジェクトのオープン操作は失敗する。オブジェクトを共有するには、次の値のいずれかを（または組み合わせを）指定する。

値	意味
FILE_SHARE_DELETE	Windows NT のみ ： 後続のオープン操作で削除アクセスが要求された場合、そのオープンを許可する。
FILE_SHARE_READ	後続のオープン操作で読み取りアクセスが要求された場合、そのオープンを許可する。
FILE_SHARE_WRITE	後続のオープン操作で書き込みアクセスが要求された場合、そのオープンを許可する。

LpSecurityAttributes：取得したハンドルの子プロセスへの継承を許可するかどうかを決める、SECURITY_ATTRIBUTES 構造体へのポインタを指定する。ここでは、NULL を指定したので、ハンドルは継承されない。

DwCreationDisposition：ファイルが存在するとき、または存在しないときのそれぞれの(＊ファイルへの) 動作を指定する。指定した OPEN_EXISTING はファイルをオー

プンし、指定ファイルが存在していない場合、関数は失敗する。デバイス (コンソールも含む) 上で **CreateFile** 関数を使う場合は、このフラグを指定しなければならない。

DwFlagsAndAttributes：ファイルの属性およびフラグを指定する。指定した FILE_ATTRIBUTE_NORMAL は特に属性がなく単独で指定する。フラグとしていくつかあるが、今回は NONOVERLAPPED モードでポートを開いているので、何も指定してない。FILE_FLAG_OVERLAPPED を書くと、時間のかかる処理に対して ERROR_IO_PENDING を返すようになる。処理が終了すると、イベントはシグナル状態に設定される。このフラグを指定したときは、ReadFile 関数や WriteFile 関数で **OVERLAPPED** 構造体を指定しなければならない。

HTemplateFile：テンプレートファイルへの GENERIC_READ アクセスを持つハンドルを指定し、テンプレートファイルは、作成しているファイルに対して、ファイル属性と拡張属性を提供する。COM ポートはデバイスなので、これを NULL にした。

CreateFile 関数が成功すると、指定したファイルのハンドルが返され、関数が失敗すると、INVALID_HANDLE_VALUE が返される。

B) **SetupComm 関数**：BOOL 型のこの関数によって COM ポートの通信パラメータ（送受信バッファ）の設定を行っている。この関数は次の 3 つのパラーメータを持っている。

HANDLE：これは COM ポートへのハンドルを指定する。

DWORD:デバイス内部の受信バッファのサイズをバイト単位で指定する。

DWORD:デバイス内部の送信バッファのサイズをバイト単位で指定する。

この関数が失敗するとゼロが返され、成功すれば、ゼロ以外の値が返される。

C) **GetCommState 関数**：これも BOOL 型の関数で、DCB（Device Control Block）構造体を使ってデバイス（COM ポート）コントロールを挿入している。パラメータは次の 2 つである。

HANDLE：これは COM ポートへのハンドルを指定する。

LPDCB：DCB 構造体へのポインター。

DCB 構造体はたくさんのメンバーを持っており、このプログラムで使ったものだけ下に説明する。

－**BaudRate**：これはモデムなどのシリアルデータ転送の個数を表わす単位で、1 秒間に送ることができる信号の個数である。同じく転送速度を表わす言葉に bps（Bit Per Second）があり、こちらは 1 秒間に転送できるビット数を意味する。1200 bps 以下の低速なモデムでは Baudrate と bps は等しいが、2400bps のモデムでは 4 相位相変

調という方法によって 1 つの信号に 2 ビットの情報を与えているので、Baudrate で表わすと 1200 となる。

－ByteSize：シリアル通信ではデータを送るときの一般的なフォーマットは図 4－２に示す。

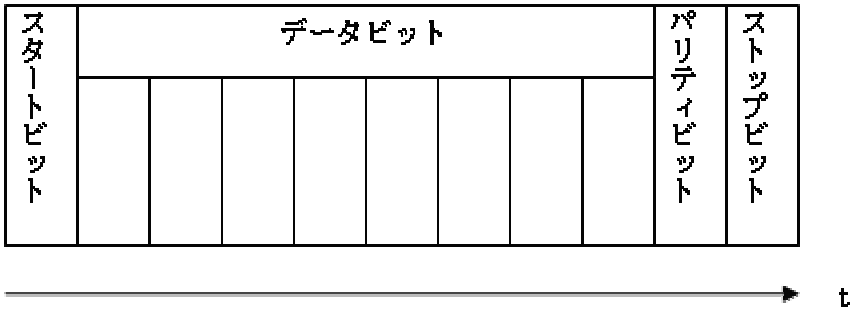


図４－２：シリアル通信で使う一般的なデータフォーマット

スタートビットは常に０で、通信データがこれから開始されると言った意味を持っている。ハードウェアはアイドル状態の通信ラインを監視している間に０を確認すると、その時点でタイミングをリセットして、一定の間隔でデータ取り込み処理が開始される。スタートビットを送信すると、それに続けて実際のデータを送信される。データは１バイトずつ送られるが、ByteSize はデータビットの長さ即ち一回に送られるデータのビット数である。これは普通、5、6、7、8 のいずれかであって、7 データビットと設定されている場合には 127 以上の ASCII コードを送受信することはできない。同じように 5 データビットの場合は 31 以上の ASCII コードは扱えない。

－Parity：スタートビットとストップビットによってデータの区切りを示すが、パリティビットと呼ばれるものを利用して、データの送信中にデータに何らかの欠落が生じていないか、それをチェックする。パリティには偶数パリティ、奇数パリティ、マークパリティ、スペースパリティ、そしてパリティ無しという選択肢がある。偶数または奇数パリティを利用すると、各データ中の 1 を数え、その数が仕様通り、偶数、または奇数になるようにパリティビットを送信する。例えば、偶数パリティを選択すると、データ中に 1 が偶数個ある場合、パリティビットは 0 となる。ですから、バイナリデータ 0110 0011 に対する偶数パリティは 0 である。逆にバイナリデータ 1101 0110 の場合、パリティビットは 1 となる。奇数パリティはこの逆で、データ中に奇数個の 1 がある場合、それは 0 になる。パリティビットによるエラーチェックは基本的な手法で、エラーが発生した時に、エラーの存在を知らせることはできるが、それがどのデータ中にあるか、その所在を知らせる機能はない。また、偶数個のエラーはデータ中で発生した場合、パリティビットでエラ

ーを検出することが不可能である。マークパリティやスペースパリティは実用的な利点がないので、現在、ほとんど利用されていません。

－StopBits:これは、シリアル通信において、ひとまとまりのデータ（データビット）の終了を通知するためのマークとして付けられるビット情報である。データの開始位置を示すビットはスタートビットと呼ばれる。ストップビットは常に１で、ビット数は、１ビット、１．５ビット、２ビット等がある。ハードウェアは、１バイト通信が終了するとストップビットを確認し、ストップビットが出力されない場合、通信が上手く行えなかったとして、フレーミングエラーを発生させる。

－ｆBainary：これは送受信されるデータのモード（Bainary,ASCII など）を決める。WIN32 API は Bainary モードしかサポートしてないので、ここではこのメンバを TRUE にしている。

－ｆOutX/ｆInX:これはそれぞれ送信と受信時に、XON/XOFF Flow Control を有効または無効に設定する。プログラムではこれらの両方を FALSE にしてフロー制御を無効にしている。

－ｆNull:これはヌルバイト（文字列の最後に自動的に付けられる文字）を捨てるか捨てないかを決める。ここでは、TRUE にしてヌルバイトを捨てるように設定した。

－fOutxCtsFlow：これは送信用のフローコントロールのための CTS (clear-to-send) シグナルをモニタルするかしないかを決める。フローコントロールを使用してないので、これを FALSE にしている。

－fRtsControl：これは RTS（Request To Send）フローコントロールのモードを設定する。ここでは、DISABLE にしている。

D) SetCommState 関数：これも GetCommState 関数と同じパラメータを持っており、GetCommState 関数で取り組んだ DCB 構造体に設定したコントロールを COM ポートに設定している。

E) SetCommTimeouts 関数：これは送信バッファに書き込むまた受信バッファから読み込むときの時間制限を設定する。次の 2 つのパラメータがある。

HANDLE: これは COM ポートへのハンドルを指定する。

LPCOMMTIMEOUTS:これはCOMMTIMEOUTS構造体へのポインタを指定する。

COMMTIMEOUTS 構造体は次のパラメータを持っている。

-ReadIntervalTimeout:これは通信線に受信される2つの文字の間に許される時間の最大値をミリ秒で表している。これは最初の文字が受信してから始まって、任意の2つの文字の間の時間がこの設定値を超えるとReadFile関数がとまってしまい、受信バッファのデータは全て返される。今回は0に設定している。

-ReadTotalTimeoutMultiplier：これは入力タイムアウトの乗数をミリ秒で指定する。毎回のReadFile関数のtotal time-outの時間を計算するために、受信バッファからReadされるバイト数にこれを掛ける。

-ReadTotalTimeoutConstant：これは入力タイムアウトの定数をミリ秒で指定する。

-WriteTotalTimeoutMultiplier：これは出力タイムアウトの乗数をミリ秒で指定する。毎回のWriteFile関数のtotal time-outの時間を計算するために、送信バッファにWriteされるバイト数にこれを掛ける。

-WriteTotalTimeoutConstant：これは出力タイムアウトの定数をミリ秒指定する。
入力タイムアウト及び出力タイムアウトの計算式は次のとおりである。

入力タイムアウト[ms]=(ReadTotalTimeoutMultiplier[ms/byte]×<入力サイズ>[byte])+ReadTotalTimeoutConstant[ms]
出力タイムアウト[ms]=(WriteTotalTimeoutMultiplier [ms/byte]×<出力サイズ>[byte])+WriteTotalTimeoutConstant[ms]

SetCommTimeouts関数が成功すれば戻り値はゼロ以外になる。

2) long APIENTRY RS_GetInBufferCount：この中には次の関数が入っている。

A) ClearCommError 関数：この関数は通信エラーが発生するときに、通信デバイスの状態を報告し、エラーをクリアするために他の Input/Output Operation を有効にする。これには 3 つのパラメータがある。

HANDLE: これは COM ポートへのハンドルを指定する。

DWORD:これは通信エラーのタイプを指定する。

COMSTAT:これは COMSTAT 構造体によって COM ポートの状態情報を取得する。

APIENTRY RS_GetInBufferCount は COMSTAT 構造体のメンバである cbInQue を使って ReadFile でまだ読み込まれてない受信バッファないのバイト数を返している。

３） long APIENTRY RS_GetOutBufferCount ： これの内容は long APIENTRY RS_GetInBufferCount と同じで、今度は送信バッファのエラー情報を報告し、エラーをクリアしている。

４）int APIENTRY RS_Write：これは BOLL 型 WriteFile 関数を使ってポートの送信バッファにデータを書き込む。バッファへのポインタの現在位置が、書き込みの開始位置になり、書き込みが終了すると、バッファへのポインタの位置は、書き込んだバイト数だけ進められる。WriteFile 関数には次のパラメータがある。

-HANDLE: これは COM ポートへのハンドルを指定する。このハンドルは、GENERIC_WRITE アクセスを持っていなければならない。

-LPSTR: これは書き込むデータが入ったバッファへのポインタを指定する。

-Long：書き込むバイト数を指定する。

-DWORD：DWORD 型の変数を指定し、この変数に、実際に書き込まれたバイト数が格納される。lpOverlapped パラメータに NULL を指定した場合は、このパラメータに NULL を指定することはできない。lpOverlapped パラメータに有効なポインタを指定した場合は、このパラメータに NULL を指定することができる。

-lpOVERLAPPED：OVERLAPPED 構造体へのポインタを指定する。COM ポートのハンドルが FILE_FLAG_OVERLAPPED フラグを持つとき、かつ、lpOverlapped パラメー

タで NULL を指定したときは、この関数は、書き込み操作が完了したと間違って通知する。FILE_FLAG_OVERLAPPED フラグを持つときは、NULL を指定してはならない。COM ポートのハンドルが FILE_FLAG_OVERLAPPED フラグを持つとき、かつ、lpOverlapped パラメータで有効なハンドルを指定したときは、OVERLAPPED 構造体で指定したオフセットからファイルが非同期的に書き込まれる。COM ポートのハンドルが FILE_FLAG_OVERLAPPED フラグを持たないとき、かつ、lpOverlapped パラメータで NULL を指定したとき（今回のプログラム）は、ファイルポインタの現在位置からファイルが同期的に書き込まれ、書き込みが完了すると、制御が返される。
COM ポートのハンドルが FILE_FLAG_OVERLAPPED フラグを持たないとき、かつ、lpOverlapped パラメータで有効なポインタを指定したときは、OVERLAPPED 構造体で指定したオフセットからファイルが同期的に書き込まれ、書き込みが完了すると、制御が返される。
WriteFile 関数が成功するとゼロ以外の値が返され、失敗したらゼロが返される。

５）int APIENTRY RS_Read：これは BOLL 型 ReadFile 関数によって、COM ポートの受信バッファからデータを読み込む。ReadFile 関数には WriteFile と同じパラメータがある。

６）int APIENTRY RS_Close：これは BOOL がた関数 CloseHandle を呼び出して、オープンしているデバイス（COM ポート）をクロズする。デバイスへのハンドルだけのパラメータを持っている。関数が成功すればゼロを返し、失敗すればゼロを返す。

VC++で MFC のダイアログベースプログラムでは、C/C++言語みたいに main 文がなく、ダイアログのインプリメンテーションファイル（このプログラムでは、RS232Dlg.cpp）に COM ポート制御用のファイル（com.cpp）をインクルードして、必要なところで com.cpp の関数を呼び出す。COM ポートの番号（Comport と定義した）、Baudrate（brate と定義）,Bytesize（bsize と定義）,Stopbits（sbits と定義）及び Parity(parity と定義)の全部の値を RS232Dlg.cpp で設定している。受信機 OEM3 では brate を 57600 で OEM4 では 115200 と設定した。

4.3　測位計算プログラムについて

測位計算プログラムはC言語で書いてある研究室所有のプログラムを必要なところだけ書き換えて本プログラムに追加という形で使用した。測位プログラムに使った

全ての変数を外部変数としてあるファイルに定義した。測位計算プログラムの基本的なところについて以下説明する。

4.3.1　外部変数

外部変数は記憶クラス指定子（例えば、static, auto など）をつけずに関数の外で定義したものである。外部変数は全ての関数で共通に参照できる。外部変数が定義される以前にそれを使用したい場合、または他のモジュルでそれを使用したい場合には、その前に extern 指定子をつけて宣言する必要がある。モジュルというのは、C ではソースファイルをいくつかに分割してコンパイルする時、一つのコンパイル単位である。このように外部変数に置いて、定義と宣言の意味が違って、定義はその変数のメモリ割り当てを行うが、宣言は同じ変数を指定したがたでしようするだけであってメモリ割り当てを行わない。測位計算プログラムでは、外部変数の数が多く、また同じ変数を色々なところで使うので、一々宣言するのは面倒である。そのため、全ての外部変数を extern 指定子をつけてあるヘッダファイルに宣言し、外部変数を使うソースファイルではそのヘッダファイルをインクルードした。

4.3.2　データの読み込み

研究室のプログラムでは、ある一定時間の GPS データがファイル化されているので、そのファイルをオープンしてデータを読み込んでいた。でも今回はリアルタイムで毎秒毎に char 型配列（rs_buf[]）に入って来ているデータをそのまま読み込み、外部変数に格納している。受信機 OEM3 では、＄RGEA（衛星の擬似距離情報のデータ）及び＄SVDA（衛星の位置情報のデータ）と OEM４では　＃RANGEA（衛星の擬似距離情報のデータ）及び＃SATXYZA（衛星の位置情報のデータ）を読み込んでいる。基準局と移動局のデータを別々に読み込むために、別々名前のソースファイルを作ったが、基準局のファイルでは rcvn＝０と移動局のファイルでは rcvn=１だけの違いで全く同じコードである。

4.3.3　衛星の選択

calc_direction.cpp のソースファイルで衛星の仰角を求め、choose_sat.cpp のソースファイルで仰角（10 度)、信号強度（３２．０）を設定し、測位に使う衛星を選んでいる。

4.3.4　測位計算

仰角及び信号強度の条件に合った衛星が 4 個以上であれば、測位計算が行う。

Sokui.cpp と minver.cpp(計算に必要な逆行列を計算する)のソースファイル内の関数を呼び出して実際に位置を計算しているのは calc_pos.cpp のソースファイル。このファイルの中で、求めた緯度、経度、高さをダイアログボックスにあるエヂットボックスに表示している。その部分のコード（VC++）は次のページに示す。

```
CString lat;
lat.Format("lat=%f",t[0]);
LATEdit->SetWindowText(lat);

CString lon;
lon.Format("lon=%f",t[1]);
LONEdit->SetWindowText(lon);

CString hg;
hg.Format("hgt=%f",t[2]);
HGEdit->SetWindowText(hg);
```

DGPS 測位のために必要な補正値を計算しているのは calc_correction.cpp のソースファイルである。衛星までの真の距離を求め、これからデータから読み込んだ擬似距離及び時計誤差を引いて、各衛星ごとに補正値を計算して外部変数に格納している。この部分のコード（C 言語）は次のようである。

```
for(i=0;i<SATn[rcvn];i++){
        prn = SVn[rcvn][i];

        //真の距離算出
  trange[prn]=sqrt((SVx[rcvn][prn]-Refx[rcvn])*(SVx[rcvn][prn]-Refx[rcvn])
                +(SVy[rcvn][prn]-Refy[rcvn])*(SVy[rcvn][prn]-Refy[rcvn])
                +(SVz[rcvn][prn]-Refz[rcvn])*(SVz[rcvn][prn]-Refz[rcvn]));

        //補正値変化率を考慮しない場合
  Correction[prn]=trange[prn]- (Pr1[rcvn][prn] + SV_corrtime[rcvn][prn]);
}
```

この補正値の外部変数を使って　DGPS＝１という条件の下で、sokui.cpp ソースファイルには DGPS 測位の基本的な部分を行っている。単独測位及び DGPS 測位の基本的な計算部分をその計算を行うための条件も含めたコード（C 言語）を下に示す。

```
//単独意測位
if(Oem3==1)
      r3[i] = Pr1[rcvn][prn] + SV_corrtime[rcvn][prn] +
              Iono[rcvn][prn] + Tropo[rcvn][prn] - r2[i];

if(Oem4 == 1)
      r3[i] = Pr1[rcvn][prn] + SV_corrtime[rcvn][prn] -
              Iono[rcvn][prn] - Tropo[rcvn][prn] - r2[i];

//移動局の DGPS 測位
if(DGPS == 1){
      r3[i] = Pr1[rcvn][prn] + SV_corrtime[rcvn][prn] +
              Correction[prn] - r2[i];
}
```

4.4　プログラムの使い方及び全体の　フローチャート

4.4.1　プログラムの使い方

このプログラムでは、PC 上の通常 COM ポートからシリアルで、及び仮想 COM ポートから無線 LAN で同時にデータを取り込み、リアルタイムで DGPS 測位を行っている。だから、プログラムを実行する前に、両方の COM ポートが正常に動いているかを確認し、ポートの番号を調べておく。通常の COM ポートは元々PC についているもので、プログラムで使うポートの番号だけ調べたら良い。でも仮想 COM ポートは PC には付いてないものなので、シリアル/無線 LAN コンバータ（EZL300W）を動かし、PC の無線 LAN を接続してから、TCP-VSP を開いて仮想 COM ポートを作成し、その番号も記憶しておく。次に、移動局の受信機を PC の通常 COM ポートと RS232C（クロスケーブル）で接続し、基準局の受信機も RS232C（クロスケーブル）で EZL300W と接続すれば準備は完了。この後、プログラム上の設定を行う。基準局と移動局用のダイアログのソースファイルである RS232nDlg.cpp と RS232Dlg.cpp　を開いて、先に確認したそれぞれのポート番号、brate、bsize、sbits、parity　の値を設定する。これで初めて、プログラムを実行して DGPS 測位ができる準備が完了になった。

今、プログラムを実行すると図 4-1 のように基準局と移動局用のダイアログが

表示される。先に基準局用のダイアログで「Open　Port」と言う名前のボタンを押して指定した番号の COM ポートを開ける。それから一番上の白いエディットボックスに受信機からデータを取り込むためのコマンドを書いて「Send Command　」ボタンを押すかまたは「Enter」をクリックすれば、2 番目の大きなエディットボックスに取り込まれたデータが表示され、プログラムによってデータが読み込まれその中の GPSTIME もエディットボックスに表示される。エディットボックスに OEM3 の場合 RGEA と SVDA また OEM4 の場合は RANGEA と SATXYZA のデータが順番でそろったところで、プログラム内で測位計算が行われ、基準局の単独測位結果が緯度、経度、高さ（m）でエディットボックスに表示される。これと同時に各衛星の補正値も計算されるが、その内一つをエディットボックスに表示される。開いたポートをクロズしたいとき、「Close　Port」ボタンを押せば良い。

これと同じことを移動局用のダイアログで行うと、プログラム内で移動局の単独測位及び DGPS 測位が行われ、結果は同じく緯度、経度、高さ（m）でエディットボックスに表示される。下にデータを取り込むためのコマンドを示す。

OEM3 の場合：

RGEA データ用：log com1 rgea ontime 1

SVDA データ用：log com1 svda ontime 1

OEM4 の場合：

RANGEA データ用：log com1 rangea ontime 1

SATXYZA データ用：log com1 satxyza ontime 1

Ontime 1 は 1 秒ごとのデータが取り込まれるという意味である。

4.4.2　プログラム全体のフローチャート

プログラム全体のフローチャートを次のページの図 4-3 に示す。

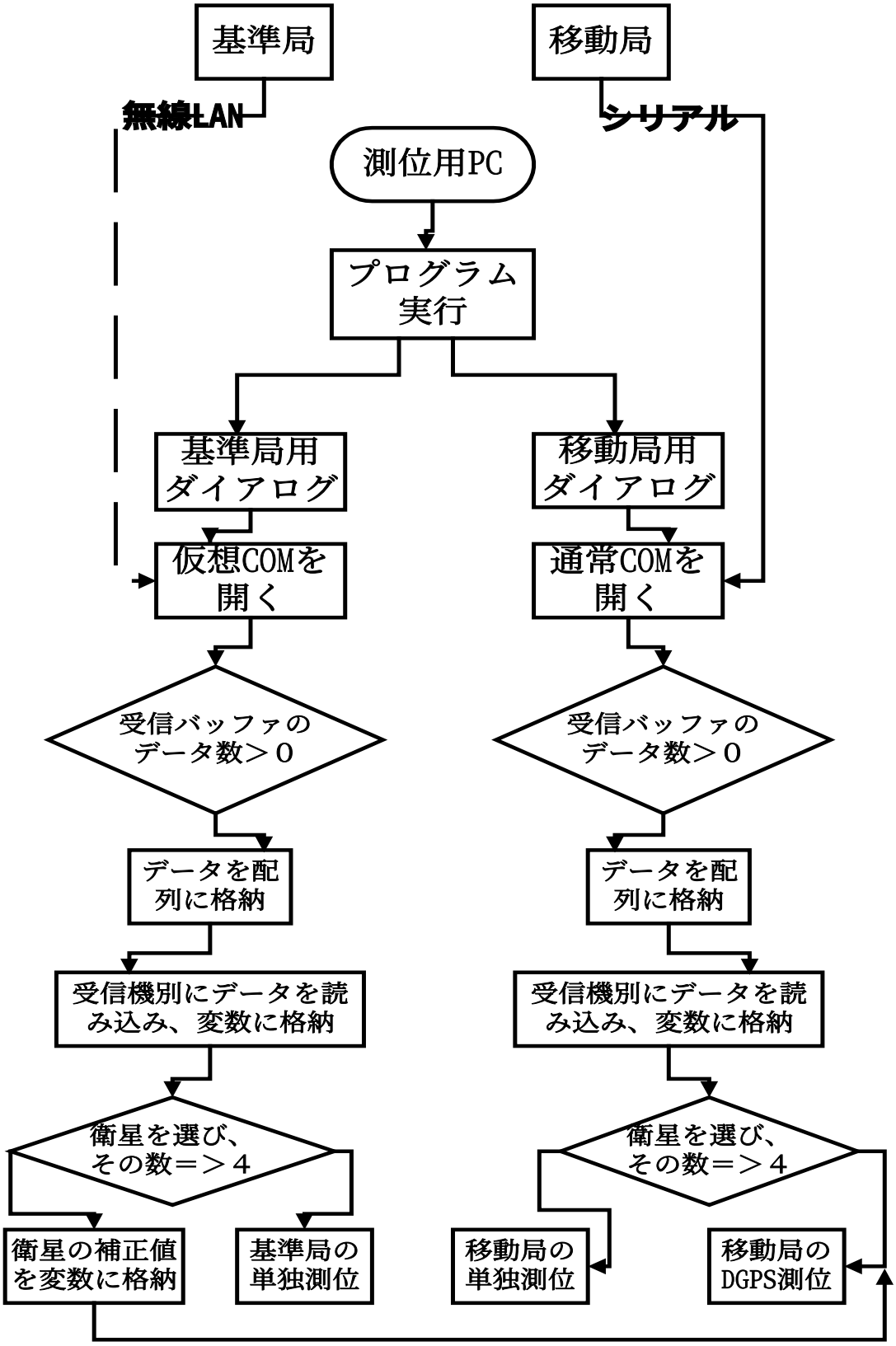


図 4-3：プログラム全体のフローチャート

第５章　無線 LAN を使った実験及びその結果

5.1　無線 LAN について

5.1.1　無線 LAN の概要

LAN は LOCAL　AREA　NETWORK の略でより対線や同軸ケーブル、光ファイバーなどを使って、同じ建物の中にコンピュータやプリンタなどを接続し、データをやり取りする（データ通信）ネットワークである。通信制御方式によって Ethernet，　FDDI（Fiber-Distributed Data Interface、光ファイバーを利用したもの）、Token.Ring などいくつかの種類がある。最も普及しているのは Ethernet 規格で、その中でも、対線を使ったスター型 LAN である 10BASE-T（最大通信速度が 10Mbps、最大伝送距離が 100m）と 100BASE-TX（最大通信速度が 100Mbps、最大伝送距離が 100m）が主流。対線とは電線を 2 本づつ撚り合わせて対にした通信用ケーブル。平行型の電線に比べてノイズの影響を抑えることができる。各ペアの周りに、雑音を遮断するシールド加工を施したものを「STP(Shielded　Twisted　Pair)ケーブル」、シールドしていないものを「UTP(UnShielded　Twisted　Pair)ケーブル」という。

無線 LAN は無線通信でデータの送受信をする LAN のことである。無線 LAN の特徴として次のものが挙げられる。

１．レイアウト変更が多いオフィスでは LAN ケーブルの引き直しの度に多くの時間と費用が費やされるが、無線 LAN ではこのような問題は生じない。

２．規格が固まったばかりで製品があまり市場に出回っていないので初期コストが高いのが欠点だが、維持コストの差を考えれば長期的には有線の LAN を用いるよりもコストが抑えられると思われる。

３．また、配線に制約の多い一般家庭でも無線 LAN は効果的で、最近ではダイアルアップルータとノートパソコンとの間を無線 LAN で結ぶ製品が広く出回っている。

４．100Mbps の 100BASE-TX 規格が浸透している有線 LAN に比べて伝送速度が 11Mbps 程度(IEEE 802.11b の場合)とさほど速くないことや、他の電気機器との電波干渉により通信エラーが発生し得ることもあり、必要に応じて有線と使い分けることが重要である。

5.1.2　無線 LAN の規格について

現在、特殊な用途を除いて、ほとんどのLANはEthernetである。Ethernet　はXerox社とDEC社(現在はHewlett Packard社の一部門)が考案したLAN規格であって、IEEE（Institute Of Electrical And Electronic Engineers、米国電気電子学会）の802.3委員会によって標準化されたものでEthernetのことをIEEE802.3とも呼ばれている。そして、IEEEの802委員会ワーキンググループ11が1998年7月に無線LANの標準規格としていくつかの規格を定めましたがこれらの規格群のことをIEEE802.11と言われる。この規格群にある無線LANのいくつかの規格を表5-１にまとめる。

表5-1：主な無線LANの標準規格

	IEEE 802.11b	IEEE 802.11a	IEEE 802.11g
周波数	2.4GHz 帯	5.2GHz 帯	2.4GHz 帯
通信速度(規格値)	最大 11Mbps	最大 54Mbps	最大 54Mbps
通信速度(実質値)	最大 5Mbps	最大 22Mbps	最大 20Mbps
伝送距離	50m～180m	20m～90m	20m～180m
互換性	高い	なし	ｂより低い
電波干渉の可能性	Bluetooth/電子レンジ	なし	Bluetooth/電子レンジ
メリット	互換性が高い	b より高速	b より高速
	安価	他の周波数帯より空いている	IEEE 802.11b と互換性がある
	比較的安定している	屋外でも使用できる	遮蔽物も透過性が高い
デメリット	速くはない	他の方式と互換性がない	他の周波数帯より混んでいる
	QoS の欠如	屋外で使用できない	比較的不安定で、高価
	WEP が脆弱	電力消費が激しい	
	同一セルに 3 台の AP を置くと、ローミングができなくなることがある	遮蔽物も透過性が低い	
		高価	

（資料提供：日商エレクトロニクス）

本研究ではIEEE802.11ｂ規格の無線LANアダプタを使用しましたので、それについて詳しく説明します。

IEEE802.11ｂでは 2.4GHｚの周波数を利用している。これは ISM（Industry Science Medical）帯域と言われ、名称通り「産業、科学、医療機器、工場生産ラインなど」の無線機器において自由に使用される周波数帯域です。なお、使用に関しては各国が独自に管理・認証を行っており、我が国の場合は無線免許なしで使用できる。

変調にはスペクトラム拡散方式を採用し、1Mbps にDBPSK（Differential Binary Phase Shift　Keying）、2Mbps にDQPSK（Differential Quaternary Phase Shift Keying）、5.5/11Mbps にCCK（Complementary Code Keying）と呼ばれる搬送波の位相差を使った１次変調を行い、DSSS（Direct Sequence Spread Spectrum 直接シーケンス・スペクトラム拡散方式）の２次変調で20MHz の帯域に拡散する。DSSSは信号にPN（疑似ノイズ）符号系列を掛け合わせ信号を拡散する方式であり、高速通信に用いらます。この方式は拡散に使用した同一の信号を使用しなければ復元できず、隠匿性と干渉に強く、現在多くの無線LANベンダはDSSS方式を採用します。これに対してFHSS(Frequency-hopping spread-spectrum：周波数ホッピング方式)は搬送波の周波数を切り替えて通信する方法であり、狭帯域での搬送に適します。

5.1.3　無線LAN（IEEE802.11ｂ）によるネットワーク構成

アクセスポイントを使うか使わないかによって、無線LANによるネットワークには次の２つのタイプがある。

1）アドホック（ad-hoc）又はピアツーピア（peer-to-peer）方式：

アドホックとは、最も基本的な無線LAN構築法であり、無線LANカードを実装したPC（無線LAN端末）間で無線通信を可能とします。一般にこの通信方式は１対１（peer-to-peer）のアドホック・ネットワークと呼ばれます。この通信方式ではネットワーク管理者の専任化あるいは格別のネットワーク構築なしで各無線LAN端末から他無線LAN端末へリソースのアクセスが行えます。アドホック無線LANは小規模な部門内通信あるいはSOHOの用途に適します。

Ethernet では、１つの通信経路を共有するために、CSMA/CD（Carrier Sense MultipleAccess/Collision Detection）というアクセス制御方式が用いられているが、無線LAN では離れれば弱まるという電波の性質上、送信中のノードが衝突検出を行うことはできない。そこで、受信側のACK 応答（ACK［ACKnowledge：肯定応答］）有無によ

って衝突を回避するCSMA/CA6with ACK（Carrier Sense Multiple Access/Collision Avoidance with ACK）というアクセス制御方式が用いられている。
また、無線 LAN 端末間において相互の距離が電波の届く範囲外にある端末による通信の割り込み（隠れ端末問題）を回避するために、送信に先駆けてこれから送信することを告げる「RTS（送信要求）」と、それに応ずる「CTS（送信可）」を送り合う RTS/CTS（Request To Send/ClearTo Send）というアクセス方法が用意されている。

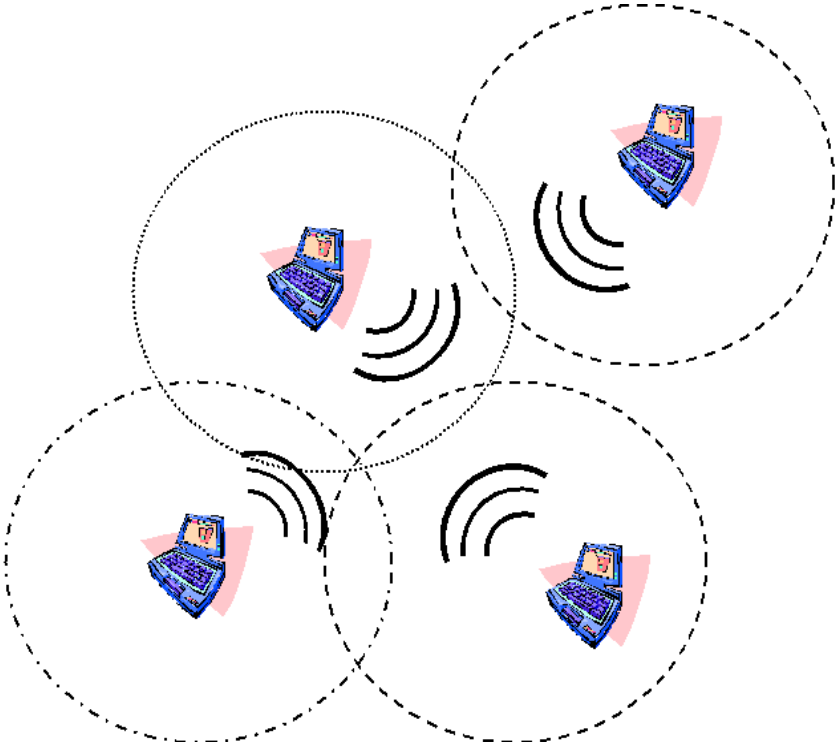


図5-1アドホックモード

2）インフラストラクチャ（Infrastructure　Mode）方式：

アドホックモードでは、互いのノードが電波到達範囲内に存在することを前提として通信が確立する。全ての端末が相互に通信するなら、サービスエリアの範囲内に全てを配置しなければならず、無線LAN の運用は大変制約の多いものとなる。そこで、携帯電話やPHS の基地局と同じようにアクセスポイントを配置し、アクセスポイント間を、Ethernet 等を使って結ぶ運用形態が用意された。これを、インフラストラクチャモードという。アクセスポイントは通常、サービスエリア内の端末が外部と通信できるように、内外の中継を行うブリッジとして機能する場合が多く、また携帯電話やPHS と

同様に、機器の仕様によっては移動中の端末が各アクセスポイントのサービスエリア間を渡り歩く、ローミング（ローミングとは、無線搭載の携帯コンピュータの利用ユーザが１つのアクセスポイントの適用範囲外に移動しても通信状態の一番良好な他のアクセスポイントに自動的に切り替え通信を可能とする機能です。）も可能な仕組みとなっている。

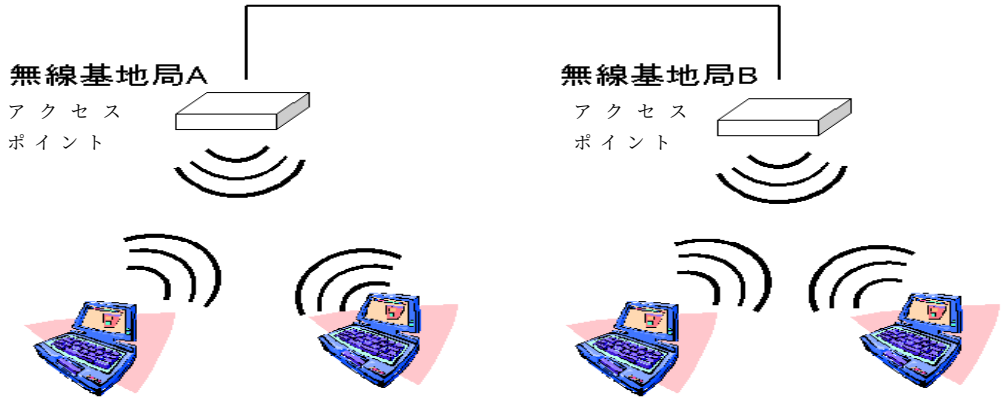


図 5-2：インフラストラクチャ方式

本研究では、アドホック方式を適用して測位計算用 PC（無線 LAN が使える）とシリアル無線 LAN コンバータの間で通信ネットワークを構成している。

5.1.4　シリアルインターフェース/無線 LAN コンバータ（EZL-300 W）

今回 GPS 受信機（OEM3/OEM4 など）からのデータを無線化するためには、α プロジェクト社から購入した EZL-300W という無線 LAN コンバータを使用した。これにはＴＣＰ／ＩＰプロトコルスタックが搭載されており、シリアルインターフェースとネットワークプロトコルをインテリジェントに相互変換することができる。したがって、既存のＲＳ２３２機器に全く手を加えることなく無線ＬＡＮネットワークとして使用することができます。EZL300W を使ったアドホックモードの概要図は図 5-3 のようになる。この状態で通信を行うためには、「ＳＳＩＤ」、「ＷＥＰ」を設定する必要があり、無線ＬＡＮカード同士は同じ値を設定しなければならない。さらに、アドホックモードでは、１台がホストの役割をするため、その１台には「ＣＲＥＡＴＥ　ＳＳＩＤ」の設定が必要で、他の無線ＬＡＮの「ＳＳＩＤ」は、同じ値を設定しなければなりません。チャンネルに関しては、どのチャンネルに設定してもホストとなるＥＺＬ－３００Wのチャンネルに自動的に合わせる。

1対複数

１対１

図 5-3：EZL-300W を使ったアドホック方式の概要図

EZL-300W の機能と動作中のランプの状態をそれぞれ表 5-1 と表 5-3 に示す。

表 5-2：EZL-300W についている機能

機能	詳細
無線ＬＡＮ インターフェース	ＰＣＭＣＩＡ （ＩＥＥＥ８０２．１１ｂ　無線ＬＡＮ）
ＲＳ２３２Ｃ インターフェース	調歩同期シリアル（ＥＩＡ－５７４準拠）　１ポート　　Ｄサブ９ピン 信号線　　：ＲｘＤ、ＴｘＤ、ＣＴＳ、ＲＴＳ，ＧＮＤ 通信速度　　：1200／2400／4800／9600／14.4K／19.2K／38.4K／57.6K／115.2K bps データビット：７／８ビット　スタートビット：１　ストップビット１ パリティ：なし／ＥＶＥＮ／ＯＤＤ フロー制御　：なし or ＲＴＳ／ＣＴＳ

対応プロトコル	ＴＣＰ／ＩＰ、ＵＤＰ、ＩＣＭＰ、ＡＲＰ、ＤＨＣＰ、ＴＥＬＮＥＴ、ＥＺＣＦＧ、ＥＡＰ
セキュリティ	ＷＥＰ
同時接続数	１
消費電流	約２３０ｍＡ　※本体のみ
電源	付属ＡＣアダプタを使用　ＡＣ１００Ｖ入力　ＤＣ５Ｖ／２Ａ出力
重量	約３１０ｇ　※本体のみ
使用温度	０℃　～７０℃　結露無し

表 5-3：EZL-300W の動作状態

ＬＥＤ	色	状態
ＰＷＲ	赤	電源ＯＮで点灯
ＳＴＳ	黄	TCP/IP でリンクが確立している間は点灯 TCP/IP でリンクが確立していない間、以下のように点滅[固定 IP]　１秒毎に点滅 [DHCP]　・IP アドレスが割り当てられていない間は、１秒間に約４回点滅、１秒消灯を繰　り返す　・IP アドレスが割り当てられると１秒間に１回の点滅
ＬＩＮＫ	緑	無線ＬＡＮにリンクが確立中に点灯
ＲＸＤ	黄	ネットワーク上のパケットデータを受信すると点灯
ＴＸＤ	緑	パケットデータ送信時に点灯

SSID の設定：

SSIDはService　Set　IDの略で無線ＬＡＮがどの相手と通信するべきかを特定するために必要となる無線ＬＡＮネットワークのＩＤです。ＳＳＩＤを指定することにより、初めて通信を行うことができます。SSIDにはBSSIDとESSIDがあって、BSSIDはBase Service Set（基本サービスセット）の略であり、アドホック方式での通信領域あるいは各アクセスポイントの通信領域となります。各領域は個別のBSS ID(BSS名)が設定され、区別が図れます。なお、BSS内に含まれる無線LAN端末は同一のBSS IDを設定する。もう一つのESSIDはExtended　Service　Set　IDの略でインフラストラクチャ方式で使われる。各アクセスポイントの通信領域であるBSSをローミング領域として統合する。これによりローミングするユーザはBSS間を自由に移動することが可能となる。但し、ESSに含まれるアクセスポイントは同一のESS IDに設定する必要がある。ESS IDの別の用途に

負荷分散がある。同一場所に複数台のアクセスポイントが設置された場合、特定のアクセスポイントに無線LAN端末からのアクセスが集中することがありますが、この負荷集中を防ぐために、無線LAN端末のESS-IDを特定のアクセスポイントのESS-IDと同一にすることにより他のアクセスポイントへのアクセスを回避でき、分散が行える。SSIDの設定には大文字と小文字は区別されるため注意が必要である。

CREATE　SSIDの設定：

　アドホックモード時にホストとなる１台に新たに作成する無線ＬＡＮネットワークのＩＤである。他の無線ＬＡＮのＳＳＩＤをＣＲＥＡＴＥ　ＳＳＩＤと同じ値に設定する事でアドホックモードでの通信を行う事ができる。

チャンネルの設定:

　チャンネルは利用する周波数のことで　IEEE802.11b では周波数2.4GHz 帯の無線を利用して通信を行いますが、実際には多くの機器で14 のチャンネルに分けられており、ユーザーは1 から14 のいずれかのチャンネルの中から選択をすることができる。ＥＺＬ－３００Wでは、「０」を設定すると、無線ＬＡＮカードのデフォルト値に設定することになる。

ＷＥＰの設定：

　これはWired　Equivalent　Privacyの略で、IEEE802.11b で定められているセキュリティ機能のひとつであって設定したパスワード（ＷＥＰキー）をもとに、送受信データの暗号化を行ないます。暗号化の方法としては、暗号化なし、64bitと128bitの2種類のキー（※３）が用意されており、この数値が大きいほど暗号が複雑化することになり、セキュリティ機能が向上される。64bit なら5 文字以内、128bit では13 文字以内の半角英数字で指定を行います。

　EZL300Wの場合これらの設定は ezSerialConfig　というソフトの画面で簡単に設定することができる。この時、無線LANカードがEZL-300Wに挿入されて無い状態でEZL-300WとPCのCOMポートをRS232Cのクロスケーブルで接続、画面からCOMポートを選びREADボタンをクリックすれば、画面の内容を書き換えるようになる。設定したい項目を書いて

　WRITEボタンを押せば、設定完了です。今回はSSIDにはDENSHI2、チャンネルを０、WEPキーをなしに設定した。その設定画面は次のようである。

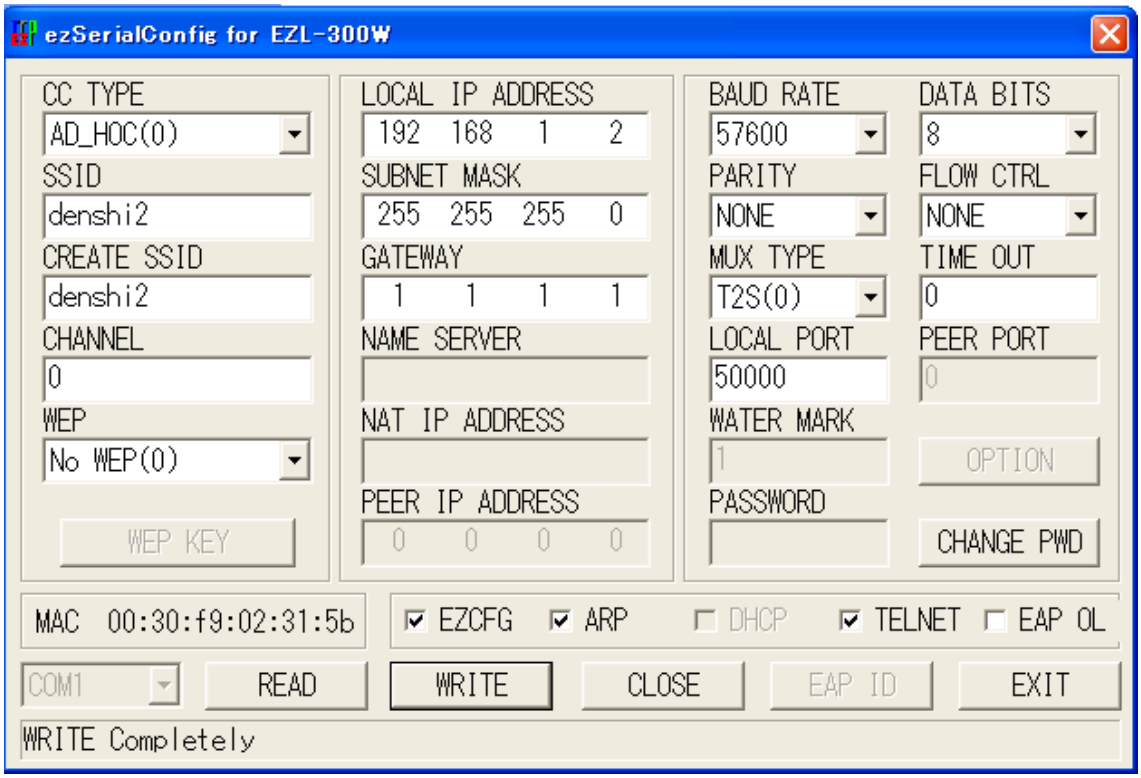


図5-4：ezSerialConfigの設定画面

MAC　アドレス：

無線 LAN ではもう一つのセキュリティ機能として MAC　ADDRESS があります。これは Media　Access　Control の略で無線アクセスポイントに接続できるパソコンを限定する。ネットワークカードに割り当てられている固有の番号で、ネットワーク機器を製造したメーカーの責任で割り当てられており、世界にただひとつしか存在しないアドレスです。マックアドレスは 12 桁構成で、前の 6 桁が製造メーカーを表し、後の 6 桁は、メーカーが割り当てた番号です。マックアドレスによるアクセス制限（ＭＡＣアドレスフィルタリング）を行うときに必要になるが、今回は１対１のアドホック方式を使ったため気にしていない。具体的に言えば、受信側（子機側）のパソコンで使用する無線ＬＡＮアダプタに固有に決められているＭＡＣアドレスを無線アクセスポイント（親機側）に記憶させて、それ以外の通信を受けつけないように設定することができる。

5.1.5　無線 LAN の伝送距離とパフォーマンス

無線 LAN（WLAN）のスピードは、有線ネットワークの効率性から建物の構造、導入する WLAN のタイプに至るまで、さまざまな要素に左右される。すべての WLAN に共通して言えるのは、WLAN のアクセスポイントとワイヤレス・クライアントとの距離が遠くなるほど、データ・スループットは低下するということである。802.11 規格は、複数のデータ伝送速度をサポートしているので、信号強度の低下に対応でき、データ・パケットの再構成にも影響を及ぼさない。WLAN クライアントが常時稼動しながら、可能な限り高い速度を検出して自動的に設定している。このため、データ伝送速度は一連の数字(802.11b の場合、伝送距離によって 11Mbps、5.5Mbps、2Mbps、1Mbps)で列挙されており、さまざまな伝送距離におけるスループットを表している。

802.11b および 802.11g が利用する周波数は、障害物を通過できるので、屋内ではたいてい最大 300 フィート(91 メートル)の伝送距離が実現している。802.11a の場合は、アクセスポイントから離れるに従ってスループットが急激に低下し、屋内における最大伝送距離はおおむね 150 フィート(46 メートル)程度とされている。伝送距離と伝送速度は、WLAN の導入環境に左右される。電波を利用した情報の送受信や伝送速度は、信号経路上にある障害物の特性に応じた影響を受ける。以下の表は、一般的な障害物による減衰の度合いを示したものである。

表 5-4：障害物による無線周波数の減衰

障害物	減衰度	例
オープン・スペース	なし	カフェテリア、中庭
木	低い	内壁、オフィスのパーティション、ドア、床
プラスタ（漆喰）	低い	内壁（古いものほど低い）
合成素材	低い	オフィスのパーティション
軽量コンクリートブロック	低い	内壁、外壁
アスベスト	低い	天井
ガラス	低い	透明ガラスの窓
ワイヤ入りガラス	中程度	ドア、パーティション
着色反射ガラス	低い	着色ガラスの窓
人体	中程度	人ごみ

水	中程度	湿った木、水槽、有機商品の在庫
レンガ	中程度	内壁、外壁、床
大理石	中程度	内壁、外壁、床
セラミック(金属含有または金属基材)	高い	セラミック・タイル、天井、床
紙	高い	ロール紙や用紙の束
コンクリート	高い	床、外壁、支柱
防弾ガラス	高い	セキュリティ・ブース
銀めっき	非常に高い	鏡
金属	非常に高い	机、オフィスのパーティション、鉄筋コンクリート、エレベーター・シャフト (昇降路)、ファイリング・キャビネット、自動消化装置、換気装置

5.2　無線 LAN の届く場所及び距離の実験

5.1.5 で説明したように無線 LAN の伝送距離は回りの環境によってよく影響される。基準局のデータを無線 LAN で飛ばすためには EZL-300W というシリアル無線 LAN コンバータを研究室に置いて、無線ＬＡＮを使えるＰＣを持ち歩いて学内のどんな場所からリンクできるかを調べて見た。実験は 2004 年 10 月 4 日午後、東京海洋大学越中島キャムパス内で行った。その実験について以下説明する。

実験結果１

実験結果を図 5-5 に示す。図中の赤いビルは情報通信研究室で、EZL-300W を北西側に置いてグラウンド側を歩いて見た。この時使った PC は無線 LAN 内蔵型の Panasonic の PC で、距離的には 200m ぐらいまで送受信ができた。でも研究室から遠くなるに連れてシグナルの強さが弱くなったり、切断したりすることが分かった。EZL-300W と PC の間にコンクリトの壁が当たると絶対に切断する。

図 5-5：グラウンド側の結果

実験結果２

実験結果を図 5-6 に示す。図中の赤いビルは情報通信研究室で、EZL-300W を実験１と逆で南東側に置いてポンド側を歩いて見た。この時 11Mbps 無線 LAN　PC カードアダプタを付けた DELL　の PC を使った。図を見ると分かるように、送受信ができた距離的は 150m ぐらいだった。またこの間でもシグナルが非常に弱くなったり、切断した

り、再接続したりなどの現象が起きた。

図 5-6：ポンド側の結果

考察

グラウンド側にはビルが少ないが、ポンド側にはビルがたくさんあるため、無線 LAN の電波が影響を受けてその届く範囲が短くなったことが分かった。また使う PC の性能及びカード型又は内蔵型の無線 LAN によってもその伝送距離が変化すると思った。でも学内の全てのところではできなくても、グラウンドまた大学院棟、図書館前の通路ではリアルタイムで DGPS 測位ができるので、色々な研究に役立つと思われる。

第 6 章　DGPS 測位実験及びシステムの評価

6.1　無線 LAN を使った DGPS 測位実験

平成 17 年 1 月 13 日の午前中は学内のグラウンドで、そして午後は大学院等前の三角形のアイランド（Island）の周り及びポンドで、本格的に DGPS 測位実験を行った。実験設備は表 6-1 の通りである。

表 6-1：実験設備

項目	基準局	移動局
位置	第 4 実験棟の屋上	グラウンド、大学院等前のアイランド、ポンド
アンテナ	NOVATEL の GPS-702	JAVAD の B-128-10
受信機	NOVATEL の OEM3	NOVATEL の OEM3
測位用 PC	なし	無線 LAN 内蔵型の NEC の PC
シリアル無線 LAN コンバータ	α-Project 社製の EZL-300W	なし
無線 LAN カードアダプタ	I-ODATA 社製の WN-B11/PCMH	なし
移動方法		台車を使用

EZL-300W の設置場所として、グラウンドで実験を行う時は研究室の向かい側の部屋の窓側に、そしてアイランドとポンドで実験を行う時は研究室奥の窓側に置いた。実験に使った基準局と移動局の外見はそれぞれ図 6-1（a）と　図 6-1（b）に示す。

図 6-1（a）：研究室内の基準局の実験風景

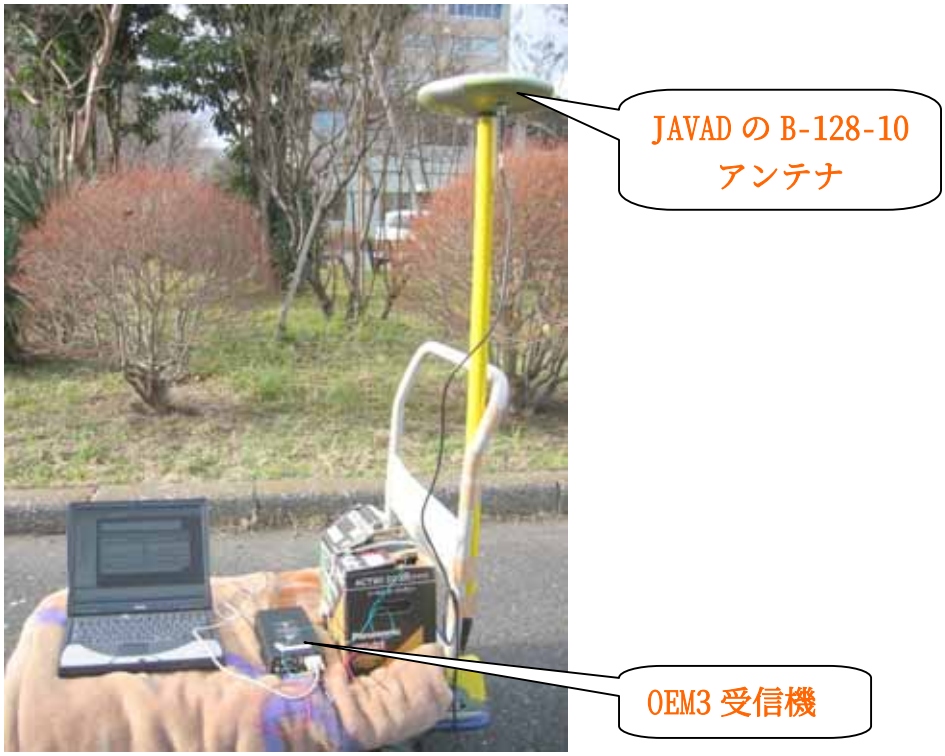


図 6-1（b）：大学院等前の移動局の実験風景

6.2 実験結果及びシステムの評価

1 月 13 日に行った実験結果をプログラムの実行画面、測位結果を学内の地図上に乗せた図を通して以下に示す。

図 6-2（a）：データ受信中の基準局用のダイアログ

図 6-2（b）：データ受信中の基準局用のダイアログ

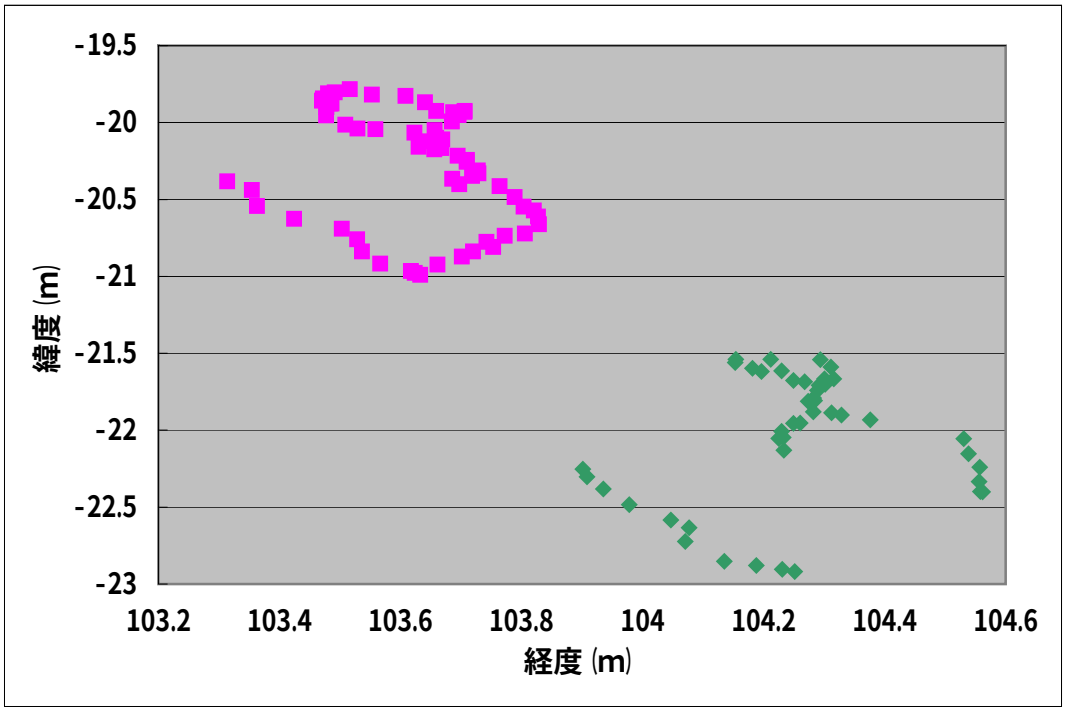


図 6-3(a)：グラウンドでの単独（ピンク）と DGPS（みどり）測位結果(X-Y 座標)

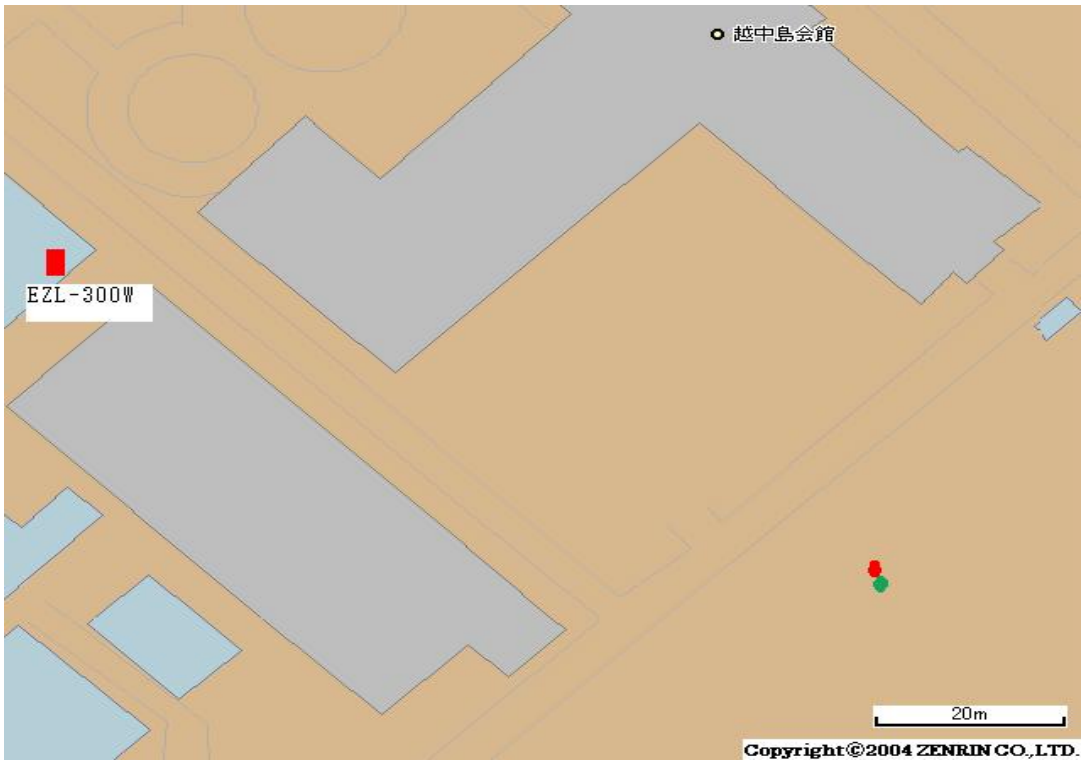


図 6-3(a)：グラウンドでの単独（赤）と DGPS（みどり）測位結果(X-Y 座標)

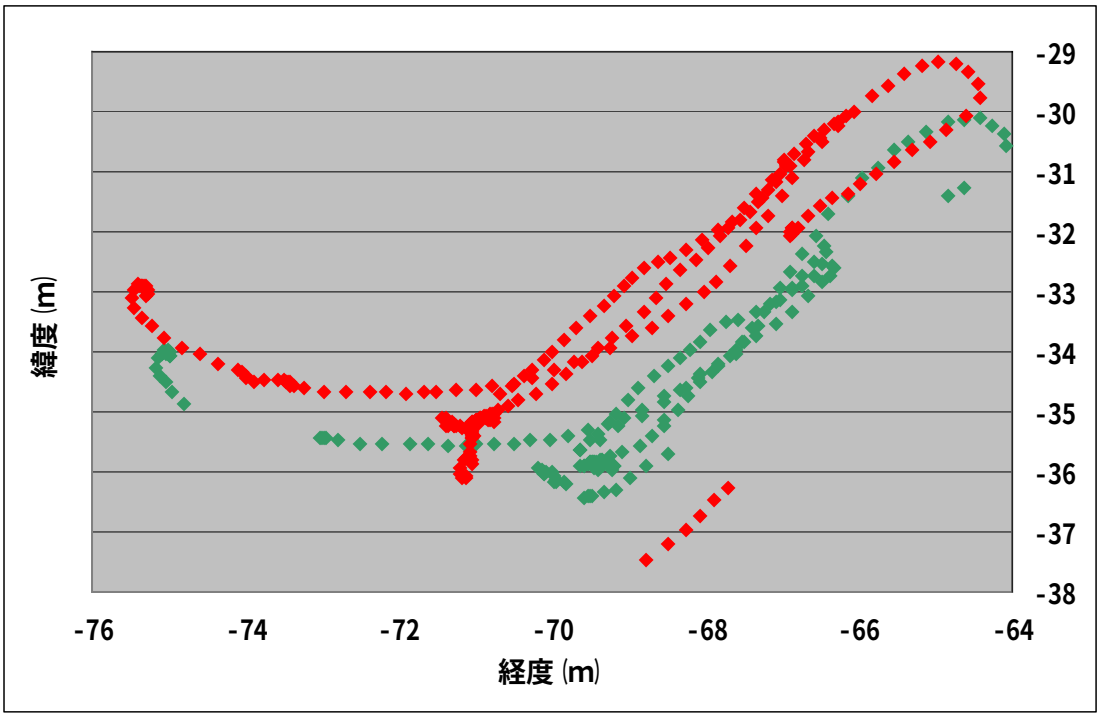


図 6-3(b)：ポンドでの単独（赤）と DGPS（みどり）測位結果(X-Y 座標)

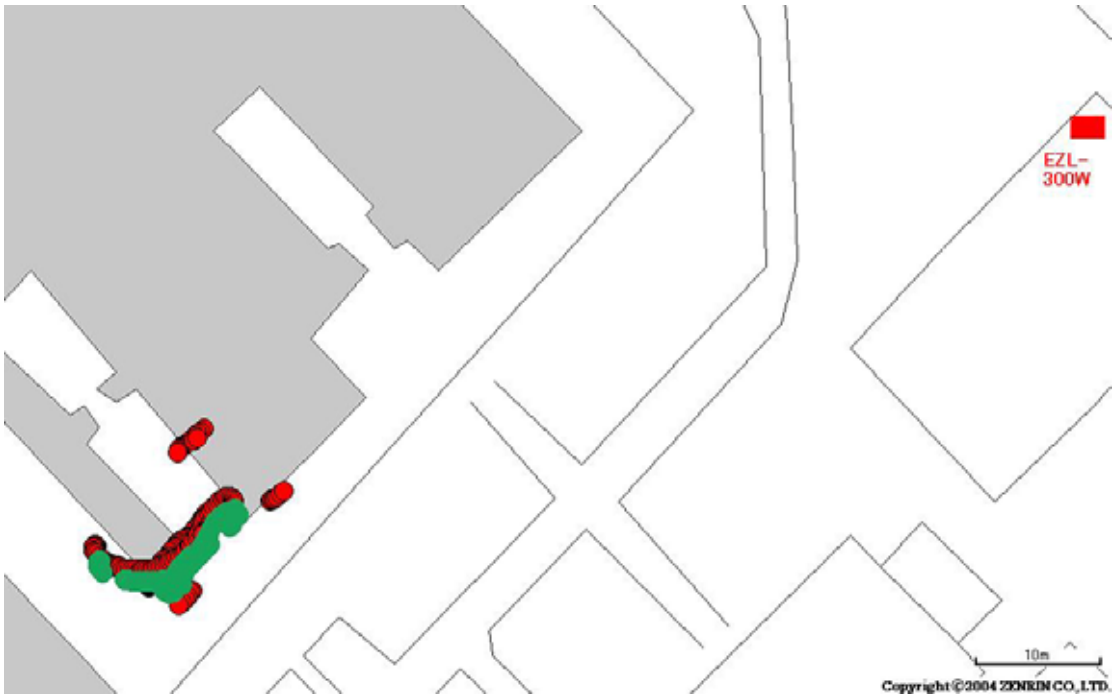


図 6-3(b)：ポンドでの単独（赤）と DGPS（みどり）測位結果(ゼンリン地図)

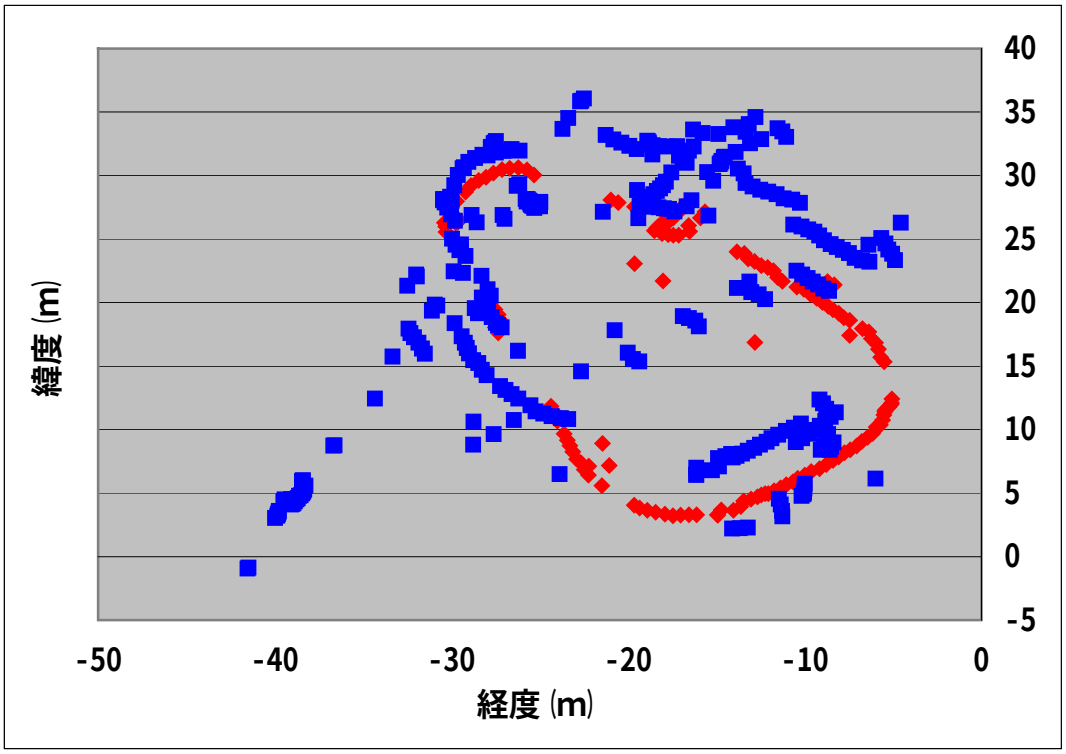


図 6-3(c)：大学院棟前での単独（青）と DGPS(赤)の測位結果(X-Y 座標)

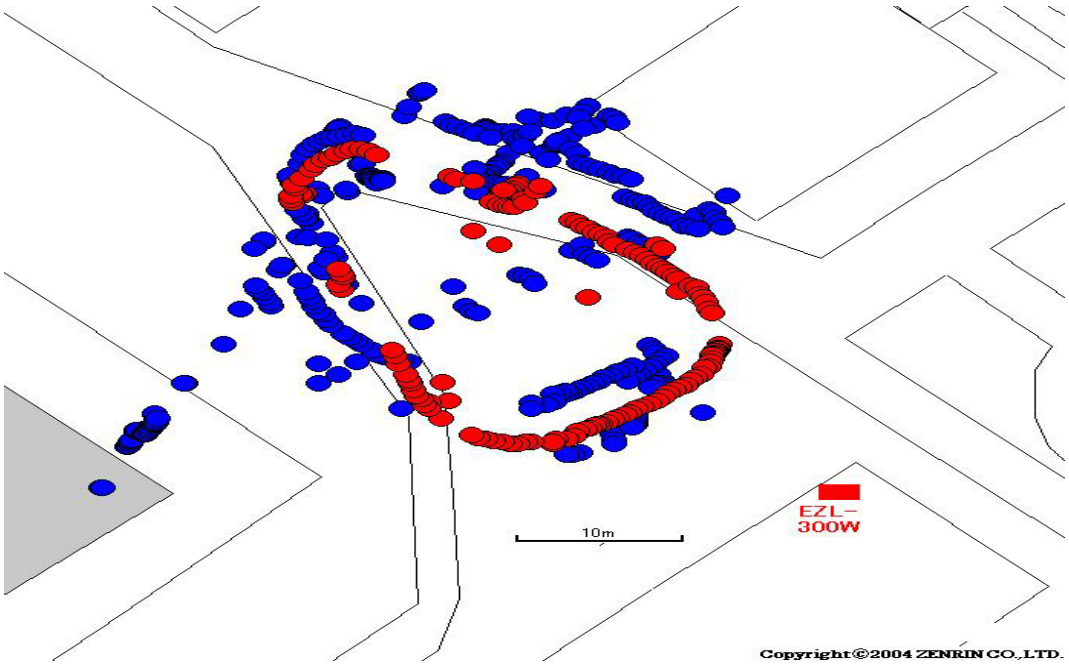


図 6-3(c)：大学院棟前での単独（青）と DGPS(赤)の測位結果(ゼンリン地図)

ゼンリン地図では10mまでしかスケールを小さくすることができない。グラウンドでは無線LANのシグナルの強さが悪くて2mぐらいの範囲しか移動してないので、ほとんど点になっている。ポンドと大学院棟前では、単独測位より、無線LANによるリアルタイムでのDGPS測位の精度がはるかによくなっていることが分かる。途中で測位が抜けているのは、データの途中切れまたデータの順番が逆になったときに測位計算ができてないためである。プログラム上のこの問題だけ解決できれば、無線LANの届く範囲ではリアルタイムでDGPS測位ができる。

第７章　結論

7.1　まとめ

第５章、無線 LAN の実験結果から、無線 LAN の送受信精度は周りの環境だけに影響されるだけでなく、PC や、内蔵型無線 LAN、PC カードアダプタ型無線 LAN のタイプに依存していることが分かった。

第 6 章の実験結果から、無線 LAN の使用が可能な環境であればリアルタイムで DGPS 測位ができ、場所によって測位精度が変化することはほとんどないと分かった。

今回作ったシステムには２つの問題がある。その一つは、プログラムが連続で 25 分までしか動かないことで、何らかの原因でプログラムが止まってしまうことである。もう一つの問題は、データが COM ポートから入ってくるとき、データの順番（RGEA と SVDA また RANGEA と SATXYZA）が途中で変わったり、先頭ではなく途中からデータが始まったりすることである。この時はデータの読み込みができなく、測位計算も止まってしまう。

幾つかの問題はまだ残っているが、リアルタイムで DGPS 測位ができたことで感動した。問題点を修正できれば、研究室のほかの研究にもこのシステムが役に立つと思っている。

7.2　今後の課題

今後の課題としては次のことがある。

（１）データの順番が変わるまたデータが途中で切れている場合でも測位計算が続けて行えるように、データの読み込みを修正し、プログラムの実用性を高める。

（２）短時間（25 分ぐらい）でプログラムが止まってしまう原因を調査し、プログラムの長時間連続使用を可能にすることで、システムの完成度を高める。

（３）RTK-GPS も基準局から補正値が必要とされ、原理は基本的に DGPS と同じであるため、今のシステムを元に、DGPS よりも精度の高い RTK-GPS 測位もできるようなシステムを作成する。

参考文献

（１）　山本信雄、「初めての MFC プログラミング」、翔泳社

（２）Beck　Zaratian、「Microsoft　VisualC＋＋6.0　プログラマーズガイド」、日経 BP ソフトプレス

（３）無線 LAN/IT 用語辞典：
http://www.keyman.or.jp/search/a_30000378_1.html?vos=nkeyadww00000001

（４）Introduction　To　RS232　Serial　Communication：
http://www.taltech.com/TALtech_web/resources/intro-sc.html

（５）Quatech/Parallel　Communication　Overview：
http://www.quatech.com/support/comm-over-parallel.php

（６）石田 剛朗、慶應義塾大学「アドホックなコミュニケーションにおける情報伝播モデルに関する研究」、修士論文2002年：
http://www.sfc.wide.ad.jp/thesis/2002/master/ishi/ishi_master-thesis20030222.pdf

（７）土屋淳.辻宏道、「GPS測量の基礎」、日本測量協会

（８）日本測位学会　編著、「GPS 人工衛星による精密測位システム」、日本測量協会

謝辞

本研究を行うにあたり、あらゆる面で多大な御指導と御鞭撻を賜った安田明生教授に厚く御礼申し上げます。

本研究に関して最初から最後まで多大な御指導とご助言をいただき、実験やプログラム作成の技術など、研究の全ての段階で有益な御示唆をいただいた久保信明助手に心より感謝申し上げます。

データ取得及び実験の際、ご協力いただいた大学院生の近藤俊一郎さんと鈴木崇史さんに心から感謝いたします。

本論文の校正やコンピュータの取り扱いにおいて、有益なご助言を賜りました博士後期課程の吉田将司さんに心から感謝いたします。

最後に、情報通信工学研究室の皆様に心より感謝いたします。