

平成11年度卒業論文

鋼プレートガーダー橋の設計及び成果品作成支援システムの構築について

鈴木雅丈

武蔵工業大学工学部土木工学科

建設情報研究室

## 目次

1 章	はじめに	
1-1	本研究の目的	3 頁
1-2	本論文の構成	3 頁
2 章	自動製図プログラムの構築	
2-1	自動製図プログラムの概要	4 頁
2-2	自動製図プログラムの使い方	5 頁
3 章	Visual Basic6 について	
3-1	Visual Basic6 の概要	6 頁
3-2	Visual Basic6 の画面構成	6 頁
3-3	Visual Basic で使う用語	8 頁
3-4	Visual Basic の特徴	10 頁
3-5	アプリケーション開発手順	11 頁
4 章	Auto CAD LT について	
4-1	Auto CAD LT の概要	14 頁
4-2	Auto CAD LT の機能説明	14 頁
4-3	座標とコマンド	22 頁
4-4	Visual Basic から AutoCADLT にコマンドを送るプログラムについて	24 頁
5 章	電子計算書プログラムの構築	
5-1	電子計算書プログラムの概要	28 頁
5-2	電子計算書の使い方	29 頁
6 章	エクセルについて	
6-1	IF 文の使いかた	30 頁
6-2	ROUND の使い方	33 頁
6-3	数式の計算演算子とは	34 頁
7 章	考察	36 頁
8 章	おわりに	37 頁
	謝辞	38 頁
	参考資料・参考サイト	39 頁
	付録	40 頁

## 1章 はじめに

### 1-1 本研究の目的

本研究は、鋼プレートガーター橋を対象として計算書及び製図の作成を支援し、より早く適切な成果品を得るためのシステムの構築を試みたものである。

製図については、マイクロソフト社の Visual Basic6 で新たにプログラミングし、その実行により、Auto desk 社の Auto CAD LT が自動的に作図作業を行うものである。電子計算書については、マイクロソフト社の Microsoft Excel(以下エクセルと称す)と同社の Microsoft Word97(以下ワードと称す)を用い、リンク貼りつけ機能を用いて、両者を関連付けることで設計計算と計算書作成の効率化をはかった。

### 1-2 本論文の構成

本論文は8章で構成されており、1章では本研究の目的について、2章では自動製図プログラムの構築について、3章では、Visual Basic6 について、4章では Auto CAD LT について、5章では電子計算書プログラムの構築について、6章ではエクセルについて、7章で考察、8章でおわりにとしてある。

## 2章 自動製図プログラムの構築

### 2-1 自動製図プログラムの概要

図 2-1 に製図のための情報入力画面の一例を示す。図中のボックスに数値を入れ、そのあとに作図ボタンを押すと図 2-2 に示す作図結果が自動的に得られる。中心点指示のボタンは作図位置を指定する際に選択し、その位置を入力するためのものである。この入力データを後述の電子計算書から取り込むことにより、自動製図が可能となるが、本研究では時間の都合とコードに関する知識不足のため、図 2-2 に示す主桁フランジの輪郭部分(上から見た図)を例として作成するのみとした。

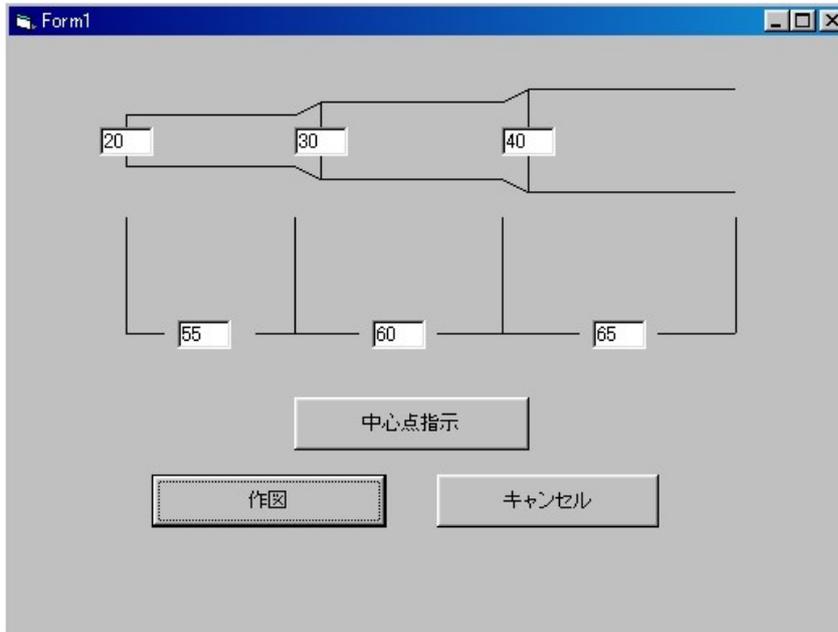


図 2-1. 入力フォーム

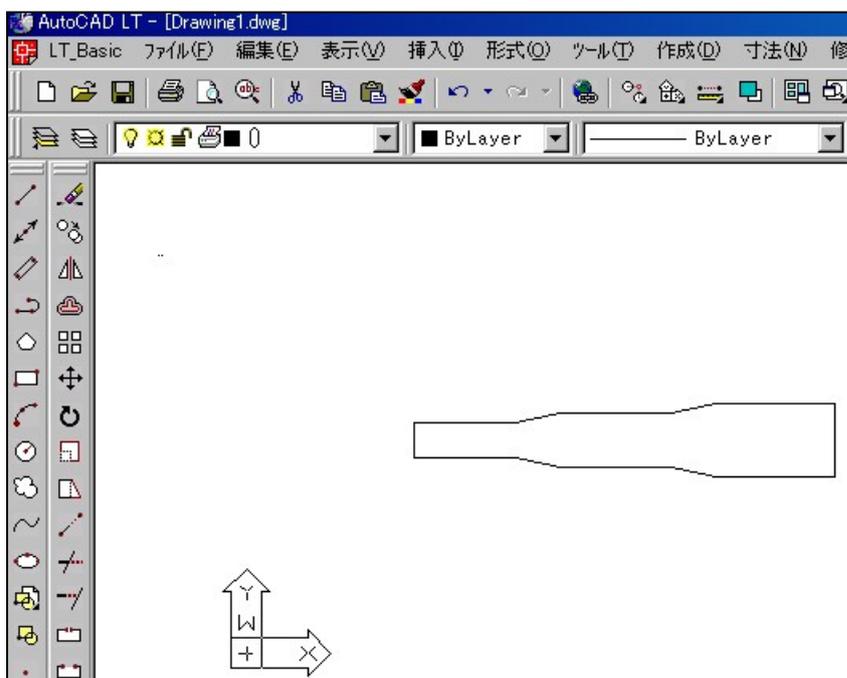


図 2-2. 自動作図画面

## 2-2 自動製図プログラムの使い方

最初に Visual Basic6 を起動させ、次に Auto CAD LT も起動させる。そして、Visual Basic6 でファイル『自動設計プログラム』を開きデバックというツールの下にあるのボタン  を押す。すると、**図 2-3** が出てくるのでその画面のテキストボックスに数値を入力する。

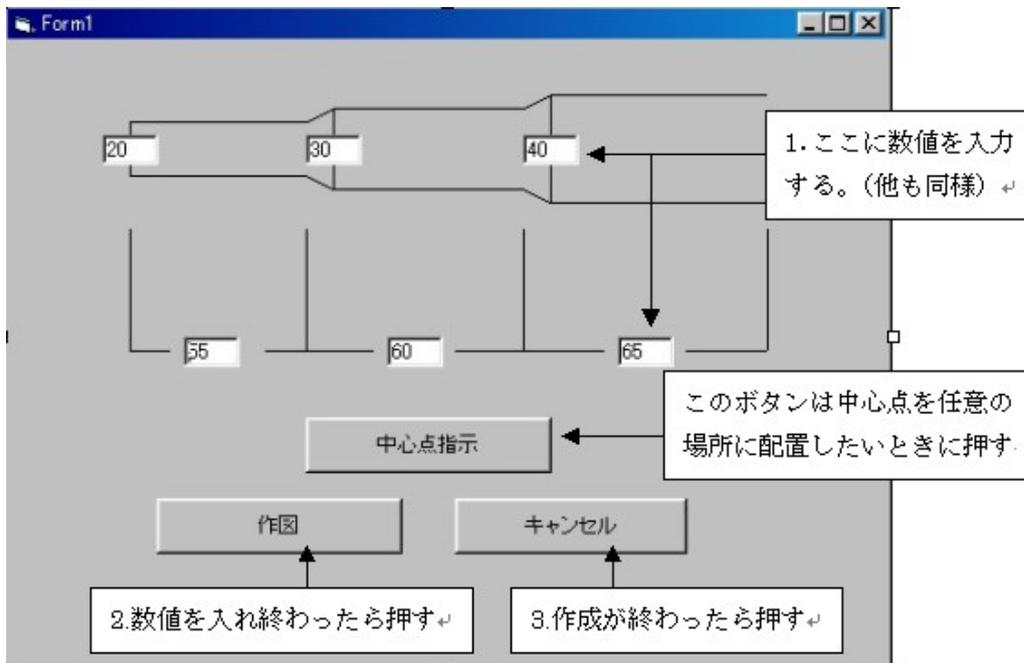


図 2-3. 自動製図プログラム入力画面

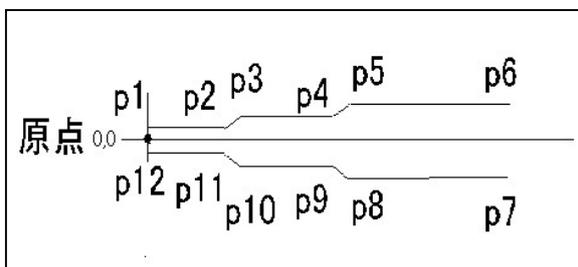


図 2-4. プログラムのイメージ

以上が自動製図プログラムの成果品の使い方である。

次に、自動製図プログラムの構築について紹介するが、内容を説明する前に使用しているソフトについての機能と使い方について述べる。なぜなら使っているソフトが使えなければ編集ができないからである。

中心点（原点）について

**図 2-4** に示す場所を原点とした。

以上の手順で操作をすると Auto CAD LT の画面に設定した値の図が作成される。

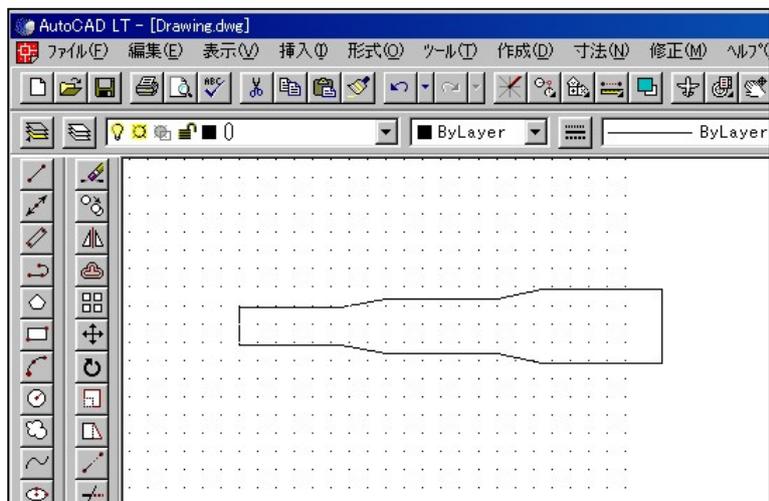


図 2-5. 自動製図プログラム実行結果

### 3章 Visual Basic6 について

#### 3-1 Visual Basic6 の概要

Visual Basic とは米マイクロソフト社が開発したプログラミング言語である。画面上のフォームに、ボタンやラベル、テキストボックスなどの部品（コントロールと呼ぶ）を配置し、各コントロールが感知するイベントごとにプログラムを記述していく。アプリケーションの GUI の部分はコントロールを配置するだけで作成できるので、GUI アプリケーションを簡単に作成できる。

GUI とは「グラフィカル・ユーザー・インターフェース」の略。グラフィックスを使ってコンピューターの操作を視覚化した操作環境のこと。Windows や MacOS, OS/2 などの OS で採用されている。

#### 3-2 Visual Basic6 の画面構成

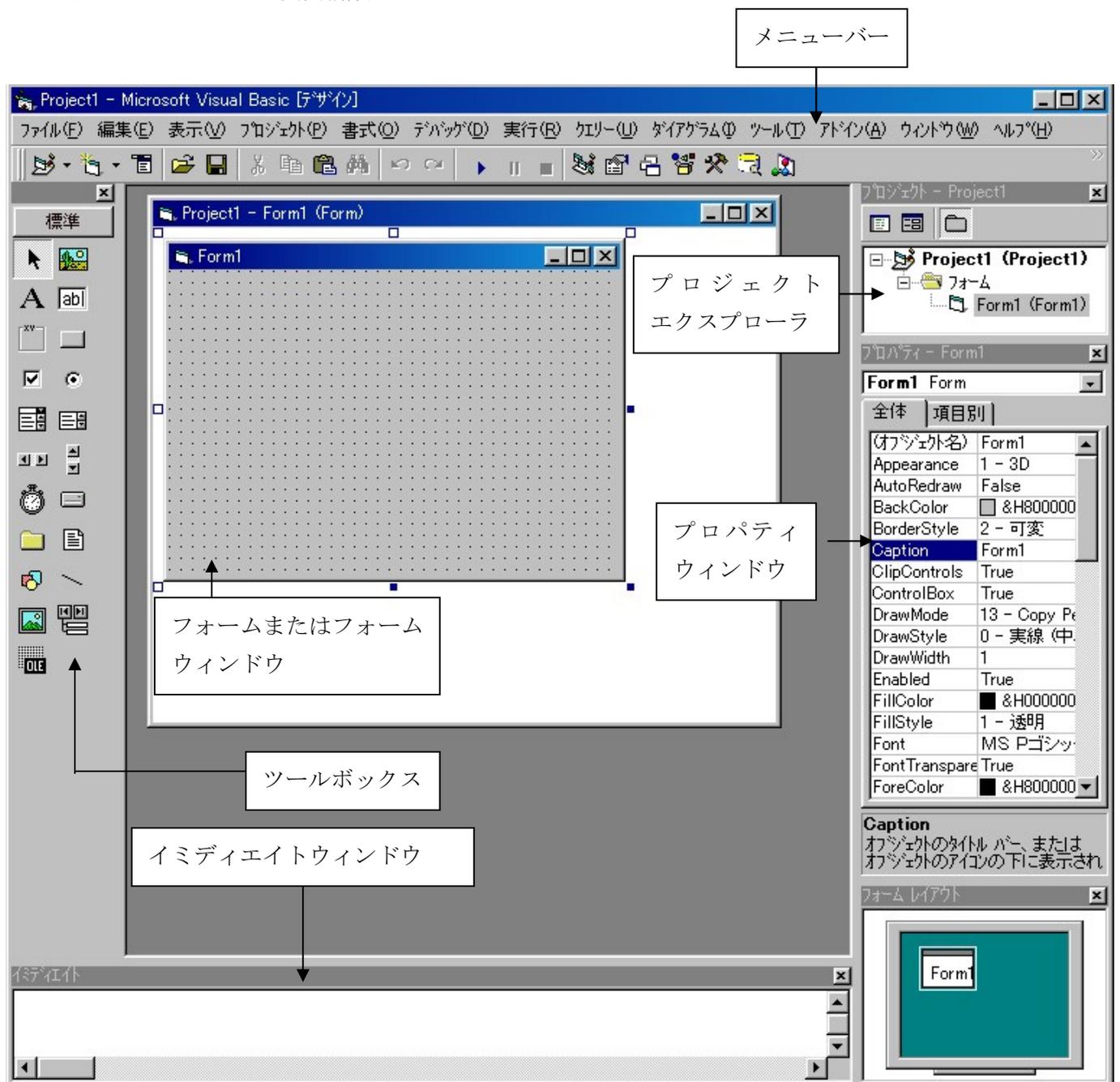


図 3-1. Visual Basic6 の画面構成

## 機能説明

### メニューバー, ツールバー

Visual Basic のメニューとツールバーです。『本書でメニューから@@@を選択して・・・』のように記述されている個所は, すべて, このメニューバーの中のメニューを実行することを意味します。

### フォーム

Visual Basic で作成するアプリケーションの中心になる画面です。Windows 上で動作するアプリケーションで例えば, 『メモ帳』も『ペイント』も, すべて特定のウィンドウ上で動作している。このようなウィンドウを, Visual Basic ではフォームと呼ぶ。アプリケーションを作成するには, フォームをデザインすることからはじめる。

### ツールボックス

フォームをデザインするためのツール集である。図は, 標準のツールボックスの内容である。『プロジェクト』 - 『コンポーネント』メニューを実行すると, このほかのコントロールを追加して利用することができる。

### プロジェクトエクスプローラ

選択されているコントロールのプロパティを設定するためのウィンドウである。プロパティは, コントロールの名前や大きさの特徴を決めるものである。

### イミディエイトウィンドウ

プログラムのデバックに使うウィンドウである。プログラムを実行したり, 実行中の値を表示したりすることができる。

### 3-3 Visual Basic で使う用語

#### アプリケーションソフト

ユーザーがある用途のために利用するソフトウェアの総称。『応用ソフト』と訳す。OS の基本ソフトウェアに対して、文書を作る、通信する、絵を書くといった目的を実現するためのソフトウェアを指す。アプリケーション・ソフトには、用途によってさまざまな種類がある。ビジネス用としてはワープロソフト、表計算ソフト、データベース・ソフト、通信ソフト、グラフィックスソフトがその一例である。また、特定業務向けに開発されたソフトや、ゲーム・ソフトもアプリケーションの一種である。

#### プロジェクト

Visual Basic で開発するアプリケーションの単位である。アプリケーション一つにつき、一つのプロジェクトを作成することになる。

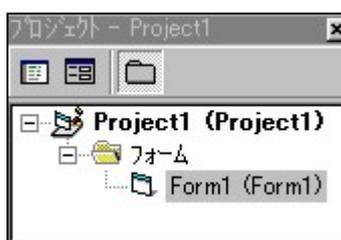


図 3-2. プロジェクトエクスプローラ

プロジェクトは、アプリケーションで使う一つ以上のファイルで構成される。プロジェクトエクスプローラを利用すると、プロジェクトに含まれるファイルの一覧を確認することができる。(図 3-2)

また、不用になったファイルを削除したり、既存のファイルを追加したりすることも可能である。

プロジェクトを保存すると、必ず “.vbp” という拡張子を持ったファイルが作成される。これをプロジェクトファイルと呼ぶ。プロジェクトファイルには、プロジェクトで使用するファイルの一覧や ActiveX オブジェクトの情報が記述されている。このファイルは Visual Basic が作成する。普段は意識する必要はない。

#### ファイル

プロジェクトを構成する 1 つの単位で、アプリケーションのデザイン時に作成するファイルに以下のものがある。

- フォームモジュール

フォームのデザインと、フォームに作成したプログラムを管理するファイルで、拡張子は “.frm” である。

- 標準モジュール

プロジェクト全体から参照するこのできるファイルである。プロジェクト全体で使用する変数やプロシージャは、標準モジュールに作成する。拡張子は “.bas” である。

- ActiveX デザイナファイル

Date Environment オブジェクトや Date Report オブジェクトなど、ActiveX デザインによって提供されるオブジェクトを管理するファイルである。拡張子は “.dsr” である。

## オブジェクト

Visual Basic のアプリケーションは、小さな集まりで構成され、この部品のことをオブジェクトと呼ぶ。フォームもオブジェクト、フォームに配置するテキストボックスやコマンドボタンのコントロールもオブジェクトである。

## コントロール

フォントを作成するときに使う部品を総称してコントロールと呼ぶ。ツールボックスには、プロジェクトで使用できるコントロールの一覧が表示される。

コントロールには、標準コントロールと、“.ocx” ファイルで提供される ActiveX コントロールの 2 種類が存在する。

## プロパティ

プロパティは、オブジェクトの名前やサイズなど、その特徴を決めるものである。

## メソッド

オブジェクトが実行できる処理のことをメソッドと呼ぶ。メソッドは、そのオブジェクトを使って何らかのアクションを実行するものである。例えば、フォームの Show メソッドを使うと、画面上にそのフォームを表示することができる。また、Hide メソッドを実行すると表示中のフォームを非表示にすることができます。

メソッドは『処理を実行する』もの、プロパティは『値を設定する』ものとする。

## イベント

Windows 上で動作するアプリケーションは、メニューが選択された、ボタンがクリックされた、Windows 上で発生するさまざまな出来事を認識し、それに応答して適切な処理を行うようにできている。例えば、Visual Basic のメニューから『実行』 - 『開始』メニューを選択すると、作成中のプロジェクトが実行される。

このように Windows 上で発生する出来事のことをイベントと呼び、イベントに応答して処理を行うアプリケーションのことをイベントアプリケーションと呼ぶ。

## イベントプロージャ

イベントプロージャは、特定のイベントに応答して実行するプログラムである。Visual Basic では、まとまりのある処理を行うプログラムの単位をプロージャと呼ぶ。

Visual Basic のオブジェクトは、発生したイベントを自動的に認識する。アプリケーションの開発者は、認識したイベントに応答してどんな処理を実行するか、その部分をプログラミングする。

## ジェネラルプロージャ

特定のイベントに割り当てられないプロージャを、ジェネラルプロージャと呼ぶ。ジェネラルプロージャは、Call ステートメントなどを使ってプログラムの中で呼び出し実行する。

### 3-4 Visual Basic の特徴

メニューから『実行』 - 『開始』メニューを選択すると、プロジェクトを実行することができる。例えば、Visual Basic を起動した直後の状態でプロジェクトを実行すると、図 1-7 のようなフォームが表示される。何もないただのフォームだが、これだけでも Windows 上で動作する立派なアプリケーションである。

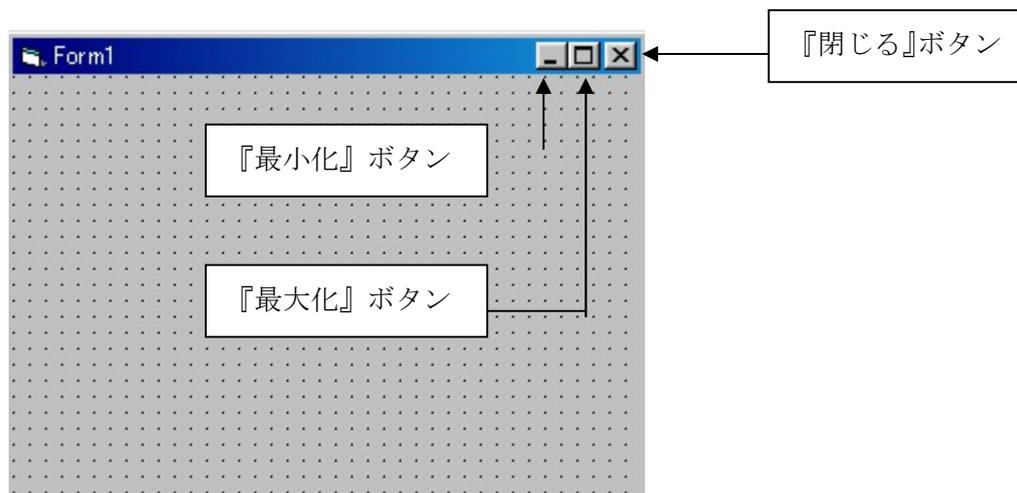
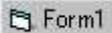


図 3-3. プログラム実行後

例をあげると、フォームのタイトルバーをつかんで（選択し、ドラックすることで）他の場所に移動したり、サイズを変更したりすることができる。また、『最小化』ボタンや『最大化』ボタン、『閉じる』ボタンも使うことが可能である。簡単に見えるが実は Windows 上では複数のイベントが発生している。フォームが移動したら、デスクトップ上の隠れている領域を再描写する処理や、『最小化』ボタンがクリックされたら、フォームをアイコン化  する処理が必要である。

Visual Basic は Windows 上で動作するために必要なこれだけの作業を、自動的に行う。よって、重要なのは『いつ、実行するか』という部分である。

### 3-5 アプリケーション開発手順

Visual Basic でアプリケーションを作成するには次の手順が一般的である。

#### 1. フォームをデザインする。

フォームは、作成したアプリケーションを操作するときの画面になります。ツールボックスに用意されているコントロールを使って、『ビジュアル』にデザインすることが可能である。

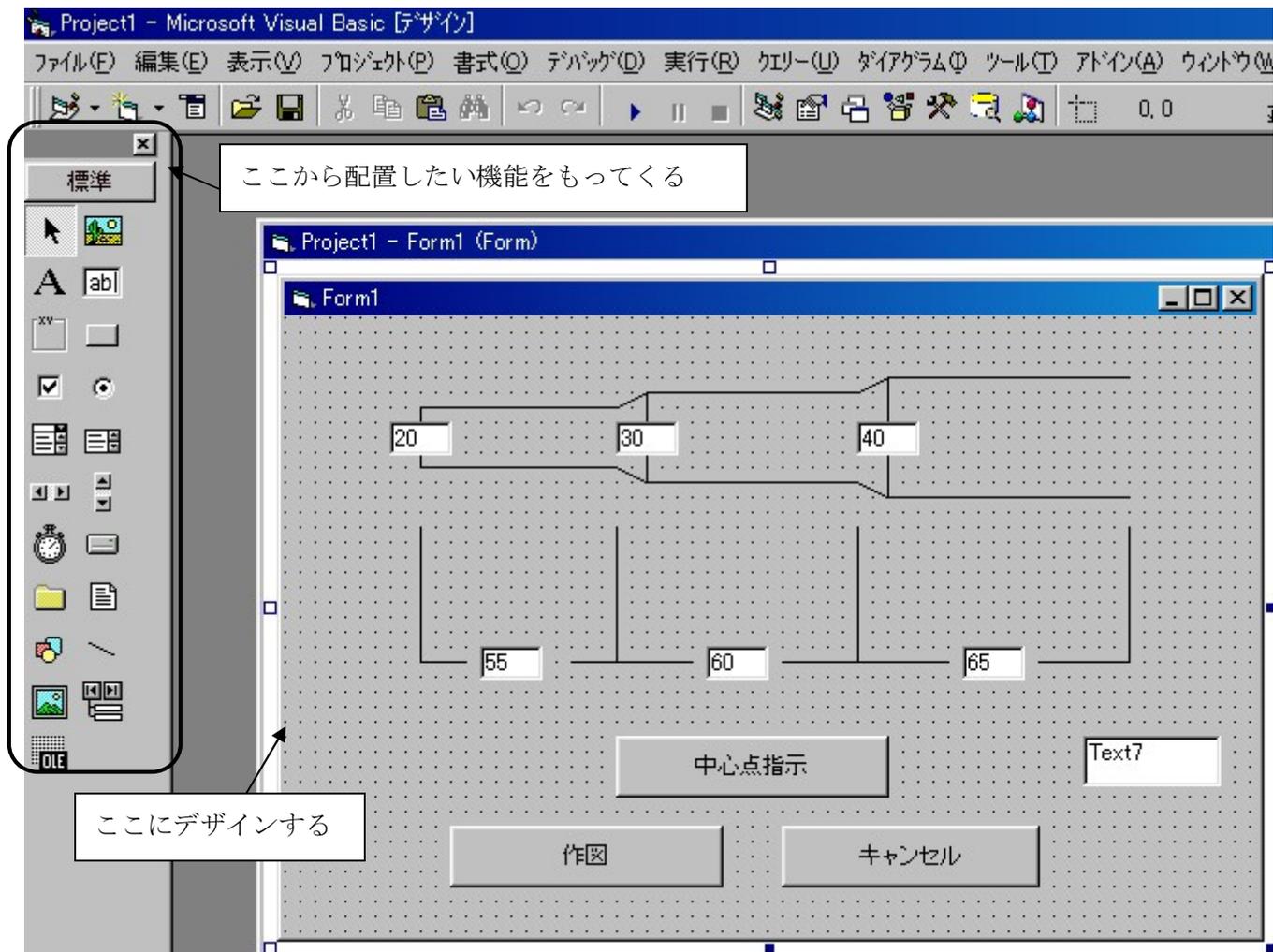


図 3-4. フォームのデザイン画面

## 2. フォームやコントロールのプロパティを設定する。

プロパティは各種コントロールの特徴を決めるものである。プロパティウィンドウを使ってデザイン時に必要なプロパティを設定しておくことで、その部分のプログラムを記述する必要がなくなる。つまり、プログラムの手順を軽減することにもなる。

プロパティの中には、オブジェクトの名前のようにデザイン時にしか設定できないものもあるので注意が必要である。

本研究ではフォームの名前の設定でのみ使用した。



図 3-5. プロパティの設定画面

### 3. プログラムを作成する。

どんなイベントが発生したときに、何を実行するか、その部分をプログラムする。



```
Project1 - Form1 (ゴト)
Command1 Click
Private Sub linkapp(Link As Control, Appname As String, Topic As String)
    Link.LinkMode = 0
    Link.LinkTopic = Appname + "|" + Topic
    Link.LinkMode = 2
End Sub
```

図 3-6. プログラム作成画面

### 4. プロジェクトを保存する。

作成したアプリケーションを保存して、いつでも利用できるようにする。プロジェクトファイルのほかにフォームモジュールなど、プロジェクトに含まれるファイルも保存する。



図 3-7. Visual Basic プログラム保存画面

## 4章 Auto CAD LTについて

### 4-1 Auto CAD LTの概要

Auto CAD LT 97とは、全世界で50万以上ものユーザーに利用されているPCベースのCADである。その実績が物語る通り、生産性に優れ、高い信頼性を誇る本格的設計・製図ソフトである。

Auto CAD LT97はローエンドのCADとして発売されている。LTという名前がついているが、操作性と互換性に優れている。より高度な機能を持つ、Auto CAD R14との互換性も100%保たれており、R14が作成したファイルを開くことはもちろん、LT97で編集後R14へ返すことも可能である。

Auto CADのファイルフォーマットの一つであるDXF形式は、CADのデータ交換として最も標準的な形式である。DXFファイル形式にするとフリーソフトであるJW-CADとのデータのやり取りも可能となる。

ただし、データの互換性は100%ではなく、情報落ちする場合もある。

### 4-2 Auto CAD LTの機能説明

Auto CAD LTの起動後の画面には、メニューバー、ステータスバー、作図ウィンドウ、そして各種ツールバーが存在する。

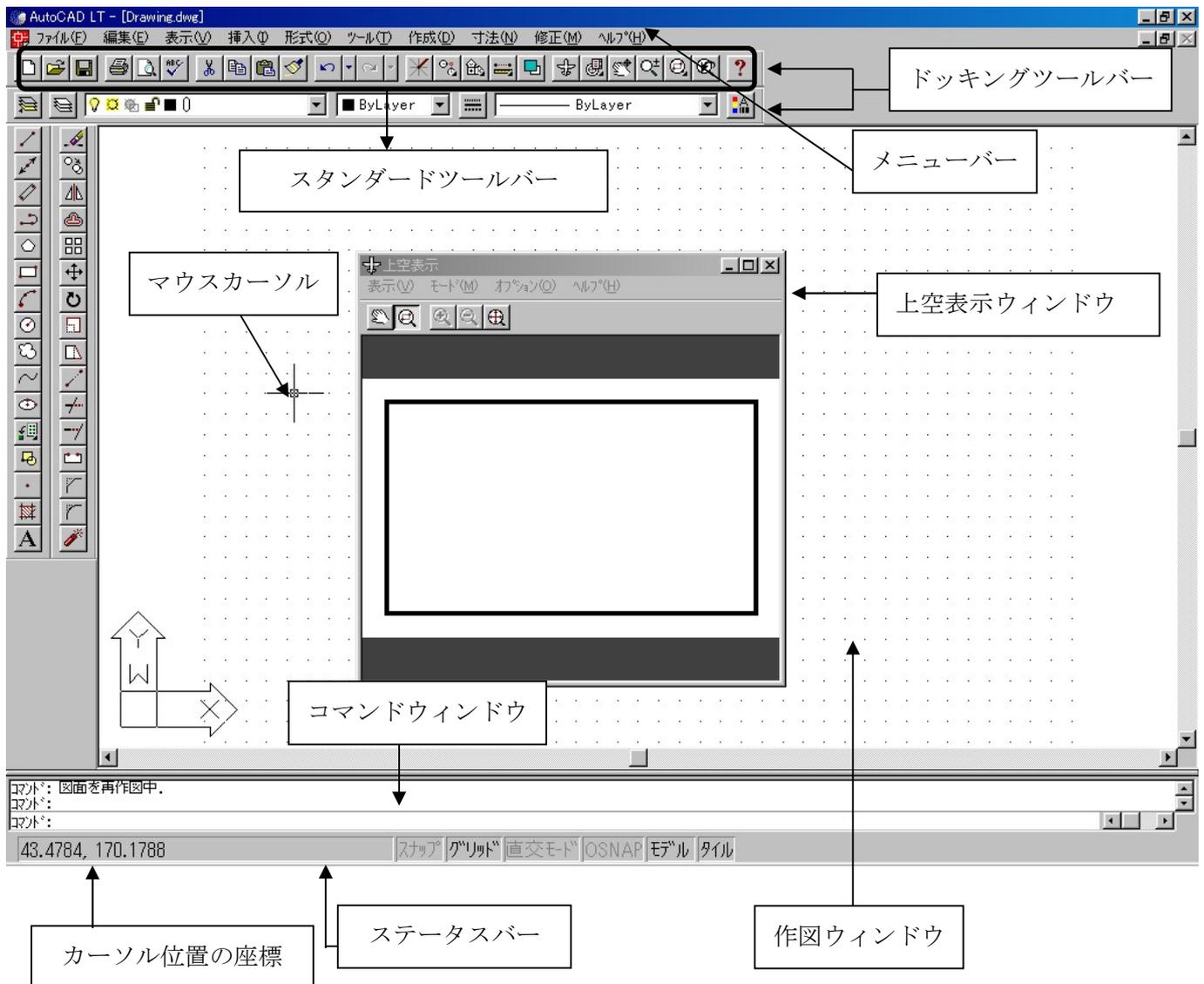


図 4-1. Auto CAD LT98 の画面構成

### メニューバー(プルダウンメニュー)

メニューバーの項目をクリアすると、プルダウンメニューが表示される。  
プルダウンメニューの項目の右側に三角形が表示される(階層メニューという)。項目の右側に『...』が表示されている場合は、その項目を選択するとダイアログボックスが表示される。

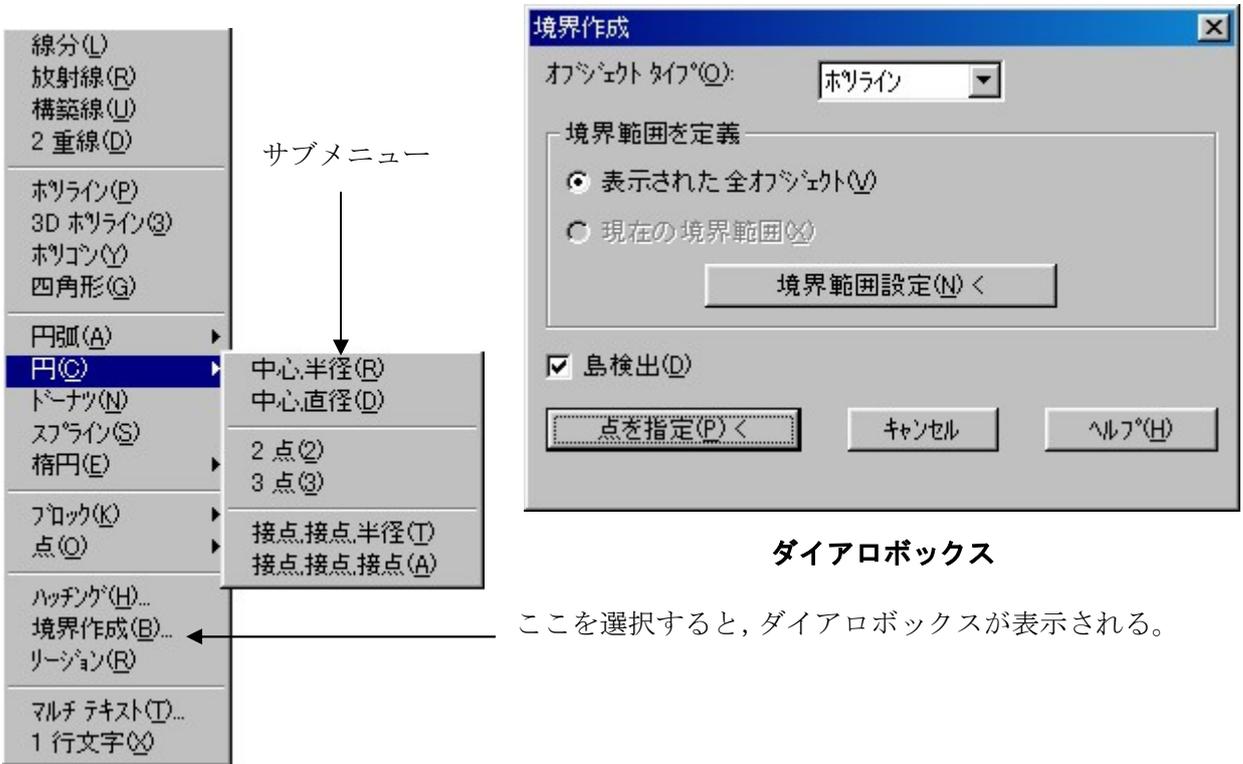


図 4-2. メニューバー

### ステータスバー

ステータスバーには、カーソル位置の座標,スナップ,グリッド,直行系モード,OSNAP(オブジェクトスナップ),モデル,タイルの現在の状態が表示される。それぞれ,ダブルクリックで ON・OFF を切り替えることができる。

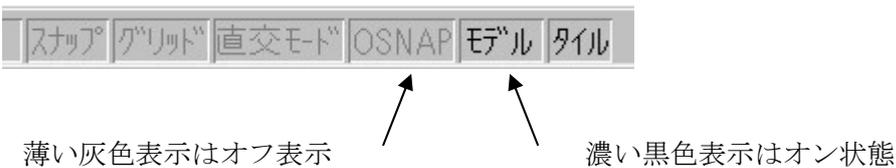


図 4-3. ステータスバー

## クイックヘルプ機能

ツールバーのアイコン上にカーソルを移動させると、そのアイコンのコマンドが表示され(ツールチップ)、ステータスバーにはその機能が表示される。

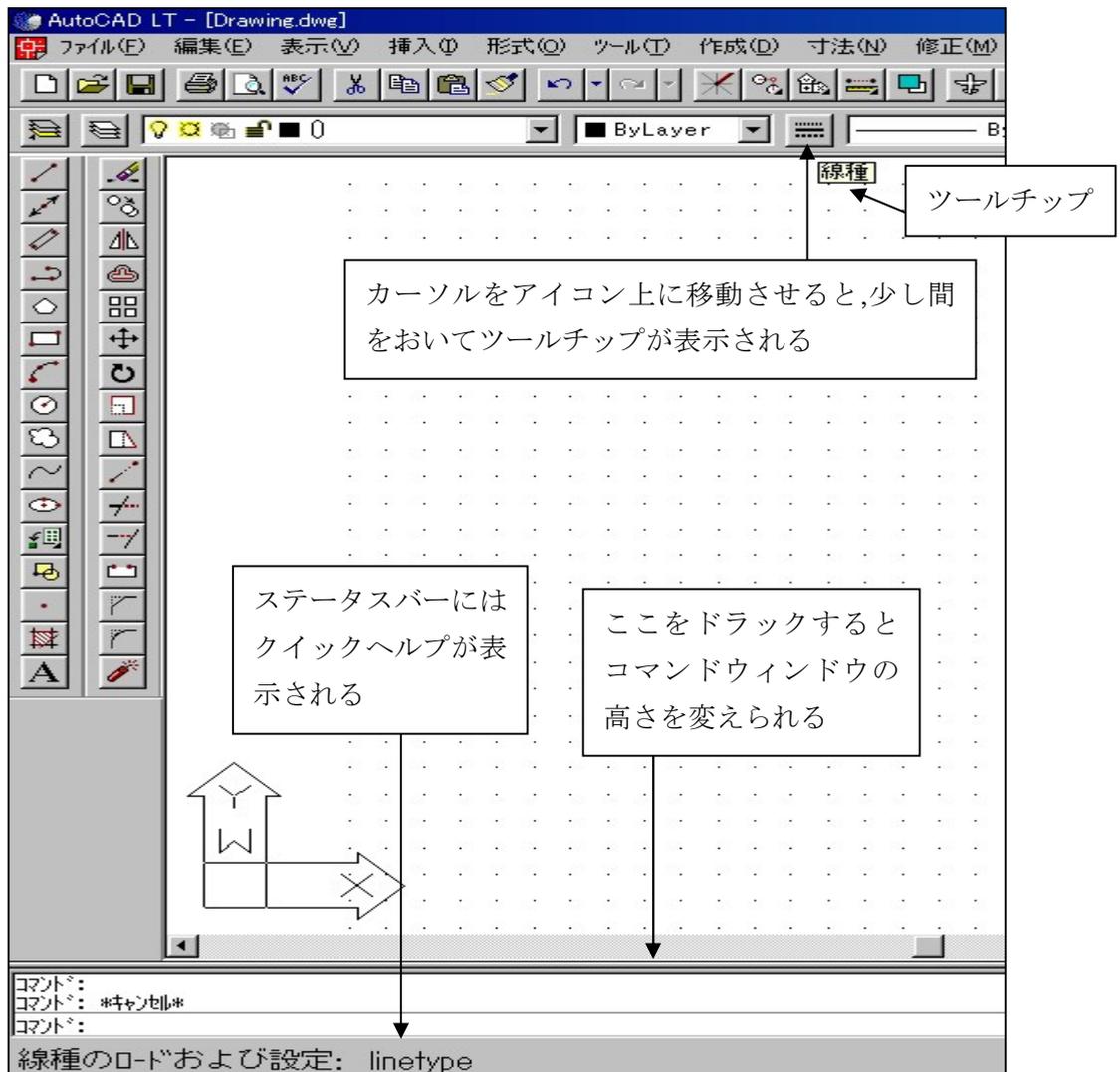


図 4-4. Auto CAD LT98 画面構成 2

## 作図ウィンドウ

作図ウィンドウは図形や文字をかいていく場所

## コマンドウィンドウ

コマンドウィンドウには、実行されているコマンド、途中経過、次の指示が表示される。指示に対してキーボードから入力した文字列も表示される。

コマンドウィンドウにもスクロールバーがあり、右スクロールバーで実行したコマンドの履歴を見ることが出来るコマンドライン上で、↑キーと↓キーを使い、コマンドラインから以前に入力したコマンドを表示し、エンターキー押すことはできる。

また、Windows の標準な編集方法(挿入、バックスペース、上書き、終了)を使ってコマンドラインを編集することが出来るので、タイプミスを効率的に修正することは可能である。

作図ウィンドウとの境界部分にマウスを動かすとマウスポインタが変わり,それをドラックすることでコマンドの高さ(表示桁数)をかえることができる。また,コマンドウィンドウも,周囲の枠部分をドラックすることにより浮動ウィンドウにすることができる。

#### テキストウィンドウ

コマンドウィンドウとは別に、『表示』メニュー→『表示設定』→『テキストウィンドウ』を選択するか、『F2 キー』を押すと,テキストウィンドウが表示される。コマンドウィンドウの履歴を表示することが可能であり,その内容は,Windows の機能であるクリップボード経由のコピーや貼り付けすることが可能なので,いろいろな活用法が考えられる。

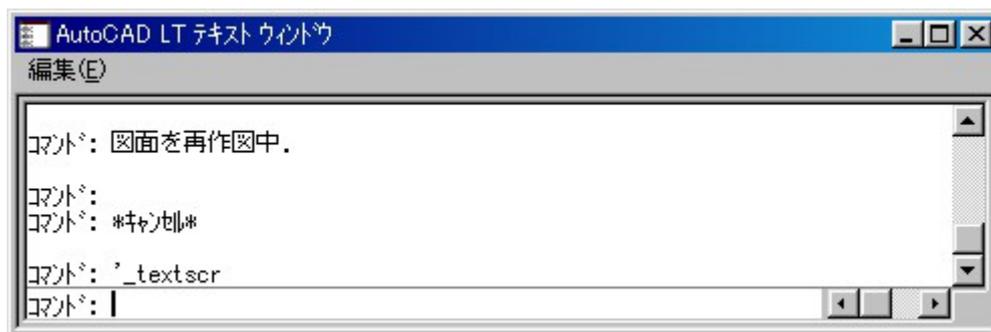


図 4-5. テキストウィンドウ

## ツールバー

ツールバーには図形の作成や編集などのためのコマンドアイコンが集められ、それぞれのアイコンをクリックすることでコマンドが選択・実行できる。メニューバーのメニューと同じ結果をワンタッチで得られる。

ツールバーにはメニューバーのすぐ下やウィンドウ枠に埋め込まれたドッキングツールと、独立したウィンドウとして配置できる浮動ツールバーが存在する。

起動画面では、『Standard Toolbar』ツールバー、『Object Properties』ツールバー、『Draw』ツールバー、『Modify』ツールバーがドッキングツールバーとして表示される。セットアップ直後では表示されないが、その数多くのツールバーが用意されている。

ドッキングツールバーの周囲の枠部分を作図ウィンドウにドラックすることで切り離し、浮動のツールバーにすることができる。逆に、浮動ツールバーをメニューの下やウィンドウ枠にドラックすることで、ドッキングツールバーにすることができる。浮動ツールバーをドッキングさせたくない場合は、『Ctrl』キーを押しながら移動させる。また、ツールバーのアイコンの右下に三角形が表示されている場合は、そのアイコンをクリックしたままにすると、さらにアイコン群がポップアップ表示されるフライアウトがある。フライアウトになっているアイコンをツールバーに配置することもできる。

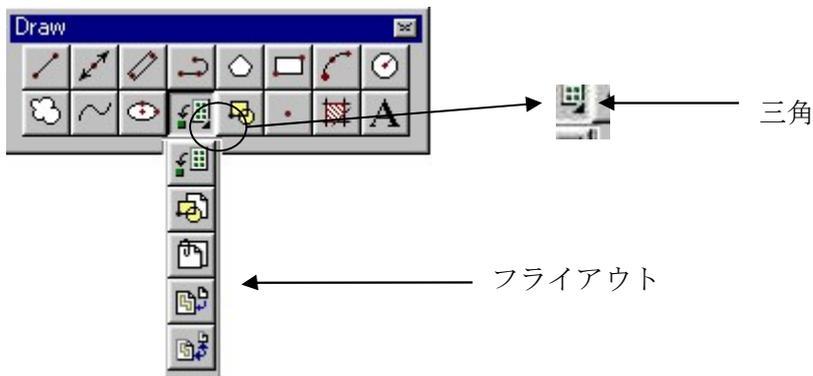


図 4-6. フライアウト

ツールチップといって、アイコン上にカーソルを置いてもう少し待つと、アイコンのコマンド名が表示される機能があり、何のコマンドアイコンであるか一目瞭然と理解できる。アイコンの表示サイズも小さいボタンと大きなボタンが選択できる。（『表示』メニュー→『ツールバー』で表示されるダイアログボックスで設定。初期設定では小さいアイコンになっている。）

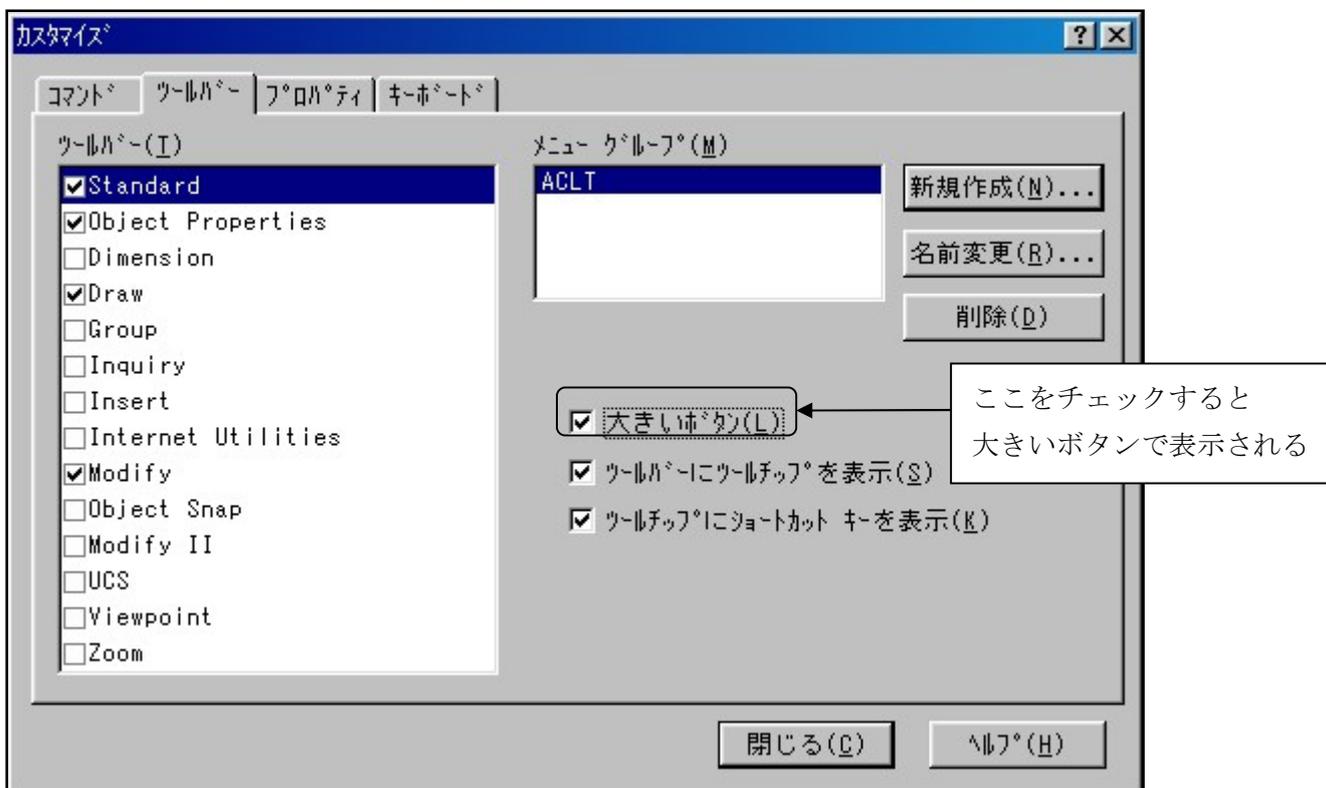


図 4-7. ツールバーのカスタマイズ

図面を作図ウィンドウとは別の視点で表示する機能として上空表示がある。上空表示は『Standard Toolbar』ツールバーの飛行機の『上空表示』  をクリックすることで実行できる。独立したウィンドウとして表示され、作図ウィンドウの表示をコントロールすることができる。

邪魔に感じるかもしれないが、なれると使いやすい機能であり、上空表示でのズームや画面の速さには定評がある。データの大ききなファイルなどでは確かに有効だ。使用するか、しないかは個人の好みとデータ量による。

上空表示

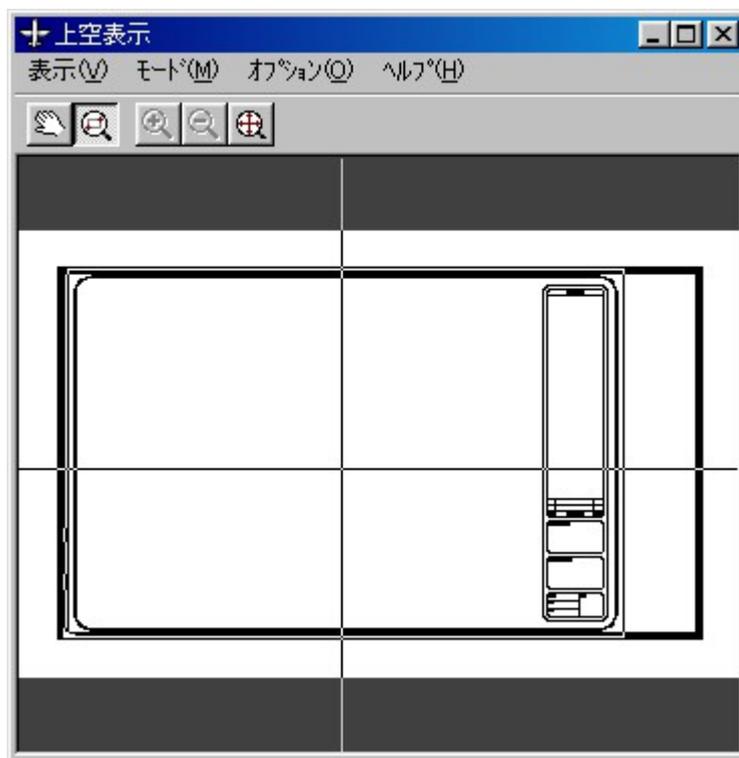


図 4-8. 上空表示画面

コンテンツエクスプローラは、図面で使うシンボルや部品をイメージで表示して簡単に使えるようにしたものである。

AutoCAD LT 98 では、部品(AutoCAD LT ではブロックという)以外にハッチングも、このコンテンツエクスプローラで表示させて利用することができる。

コンテンツエクスプローラ →



図 4-9. コンテンツエクスプローラ画面

### 4-3 座標とコマンド

では、プログラムする前に座標についての説明と、コマンドの説明をする。ここでは、直に図を書くが、後になって使うコマンドのイメージがしやすいようにするため、重要なコマンドのみ説明している。

座標で位置を決める

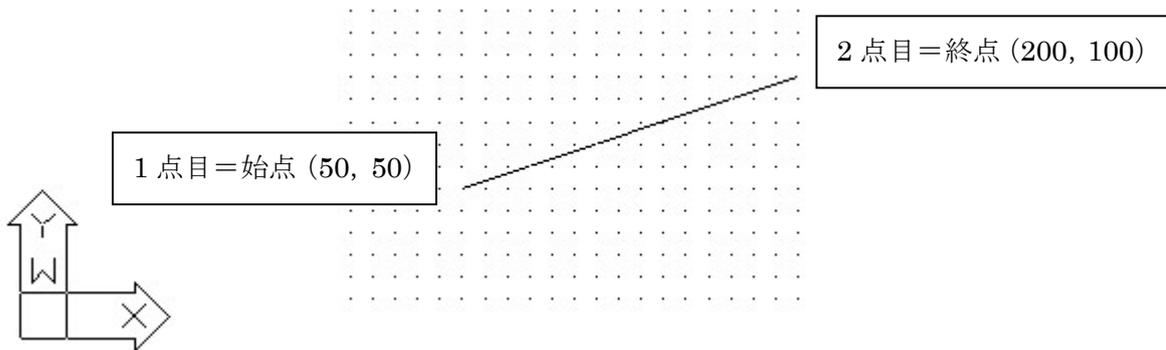
#### 絶対座標

AutoCAD LT では、ワールド座標系(WCS)という座標を使用しているが、これは絶対座標（デカルト座標）とも呼ばれる。

ワールド座標系では X,Y,Z という 3 本の軸を持っている。2 次元の図形をかき場合は XY 軸を使用し、3 次元の図形をかき場合は XYZ 軸を使用する。座標値の表し方は、原点はそれぞれ (0,0) (0,0,0) で、例えば 2 次元の場合、原点 (0,0) から X 軸方向に 110,Y 軸方向に 80 はなれた場合は (110,80) と表現する。ただし、コマンドウィンドウにキーボード入力するときは、( ) は必要ない。

実際に絶対座標で線分をかき

**図 4-10.** のように線分を絶対座標入力（始点座標値 (50,50) ,終点座標値 (200,100)）でかき。



**図 4-10.** 絶対座標

1. 『Draw』ツールバーの『線分』  をクリックする。
2. コマンドウィンドウの指示に従い、次のようにキー入力していく。  
\_line どこから : 50,50 『エンターキー』 (1 点目の指示)  
どこへ : 200,100 『エンターキー』 (2 点目の指示)  
どこへ : 『エンターキー』 (確定して終了)

以上の順で図ができあがる。

## 相対座標

相対座標に原点が (0,0) に決められているのに対して、直前に指示した点を基準点とするものが相対座標である。たとえば、先ほど絶対座標でかいた線分は絶対座標で表現すると始点 (50,50) を基準点、つまり、(0,0) と考えると、終点 (200,100) は X 軸方向に 150、Y 軸方向に 50 変化するので (150、50) と表される。

座標値の前に『@』記号をつけると、相対座標で指示したことになる。この『@』記号の有無で相対座標と区別するようになっている。

## 相対座標で線分をかく

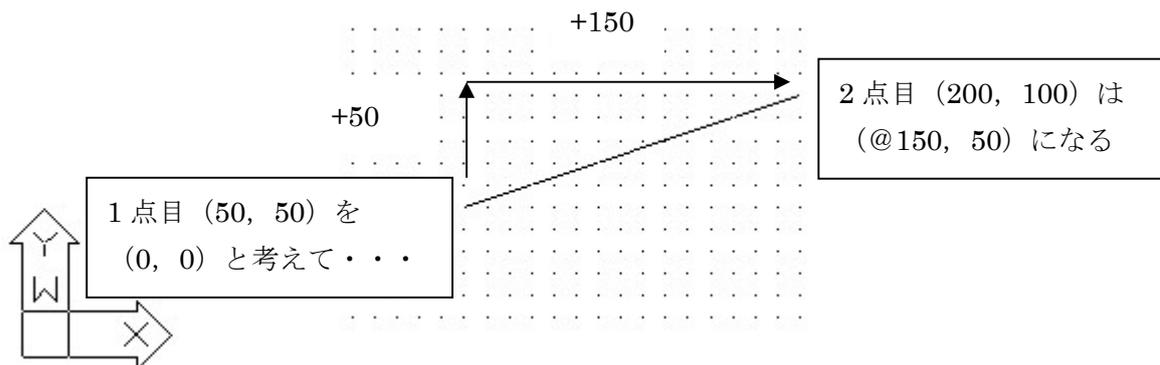


図 4-11. 相対座標

1. 『Draw』ツールバーの『線分』  をクリックする。
2. コマンドウィンドウの指示に従い、次のようにキー入力していく。

\_line どこから : 50,50 『エンターキー』 (1点目の指示)

どこへ : @150,50 『エンターキー』 (2点目の指示)

どこへ : 『エンターキー』 (確定して終了)

以上の順で図ができあがる。

その他に極座標というものがあるが、使わないのでここでは省略する。

#### 4-4 Visual Basic から AutoCADLT にコマンドを送るプログラムについて

##### H 型鋼断面図の作図プログラムと実行手順

Windows95/98 のもつ DDE (Dynamic Data Exchange) 通信の機能を使い, Visual Basic から AutoCADLT にコマンドを送ることで作図する。

##### DDE 通信の概要

複数のアプリケーション間でデータやメッセージをやり取りする、Windows のアプリケーション間通信機能。「動的データ交換」と訳す。例えば、グラフィックスソフトで作成した絵を DDE(dynamic Data Exchange)でワープロソフトの文書に取り込むと、グラフィックスソフトでもとの絵を編集することで、ワープロソフトに取り込んだ絵を更新できる。

##### DDE を使う条件

DDE を利用するには、2 つのアプリケーションが DDE に対応していることが必要。ただし Windows のアプリケーション間通信は、現在ではより高性能な OLE に引き継がれており、データのやり取りに DDE を利用することはほとんど無い。

##### プログラムの要点

Visual Basic から AutoCADLT にコマンドを送るには、まず、DDE 通信の手続きが設定されていなければならないので、DDE 通信開始手続き部分の記述が必要である。

エディットウィンドウで、『オブジェクト : (General)』『プロージャ : linkapp』に以下のように記述する。

##### <DDE 通信開始手続きの記述 1>

```
Private Sub linkapp (Link As Control, Appname As String, Topic As String)
    Link.LinkMode = 0
    Link.LinkTopic = Appname + "|" + Topic
    Link.LinkMode = 2
End Sub
```

以上が DDE 通信の手続きだが、一般的にこのプログラムをすれば手続きは完了するので普段はこれをそのままコピーして貼りつければよい。ここでは難しいことを抜きにするためコードの意味は説明しない。詳しい内容は Visual Basic のヘルプを参照。

次に、エディットウィンドウで、『オブジェクト : From』『プロージャ : load』に切り替え、次のように記述する。

##### <DDE 通信開始手続きの記述 2>

```
Private Sub Form_Load ()
    Linkapp acadtext, "AutoCAD LT.DDE", "System"
End Sub
```

『“AutoCADLT.DDE”』が、DDE 通信で使う AutoCADLT のアプリケーション名で、『“System”』がトピック名である。

## OLE (Object Linking and Embedding)

Windows 上で複数のアプリケーションを連携して利用するための仕組み。もともとはクリップボードや DDE(動的データ交換)などの機能が発展したものであったが、その後 Windows システムそのものの設計思想に関わる重要な技術となった。初期の OLE1.0 は、ある文書の中に別のアプリケーションで作成したデータを取り込んだ場合ドキュメントを作成するための機能であった。OLE1.0 の機能を利用すると、例えばワープロ部署の中にグラフィックスソフトで作成した図やイラストを取り込むことができる。

OLE1.0 でデータを取り込むには、(リンク) と (埋め込み) という 2 種類の方法がある。リンクは作成したデータを取り込んで文書の中に取り込む方法で、取り込んだデータの実体は文書の中には保存されない。一方の埋め込みは、文書中に作成した枠の中に、別のアプリケーションで作成したデータを取り込む方法。リンクも埋め込みも、取り込んだデータをダブルクリックすることで、そのデータを作成したアプリケーションが作成でき、データを編集@加工した結果はすぐに文書内に反映される。

### フォームの記述

次のようなパラメータ入力ダイアログボックス『自動製図プログラム』(フォーム=『自動製図プログラム, FAM』) を記述する。

このダイアログボックスの中に DDE 通信用のテキストボックスを作成し,Name プロパティを『acadtext』に,Visible プロパティを『False』に設定する。

このテキストボックスは,プログラム実行時には見えなくなるので,大きさや位置は邪魔にならないようならば適当でよい。

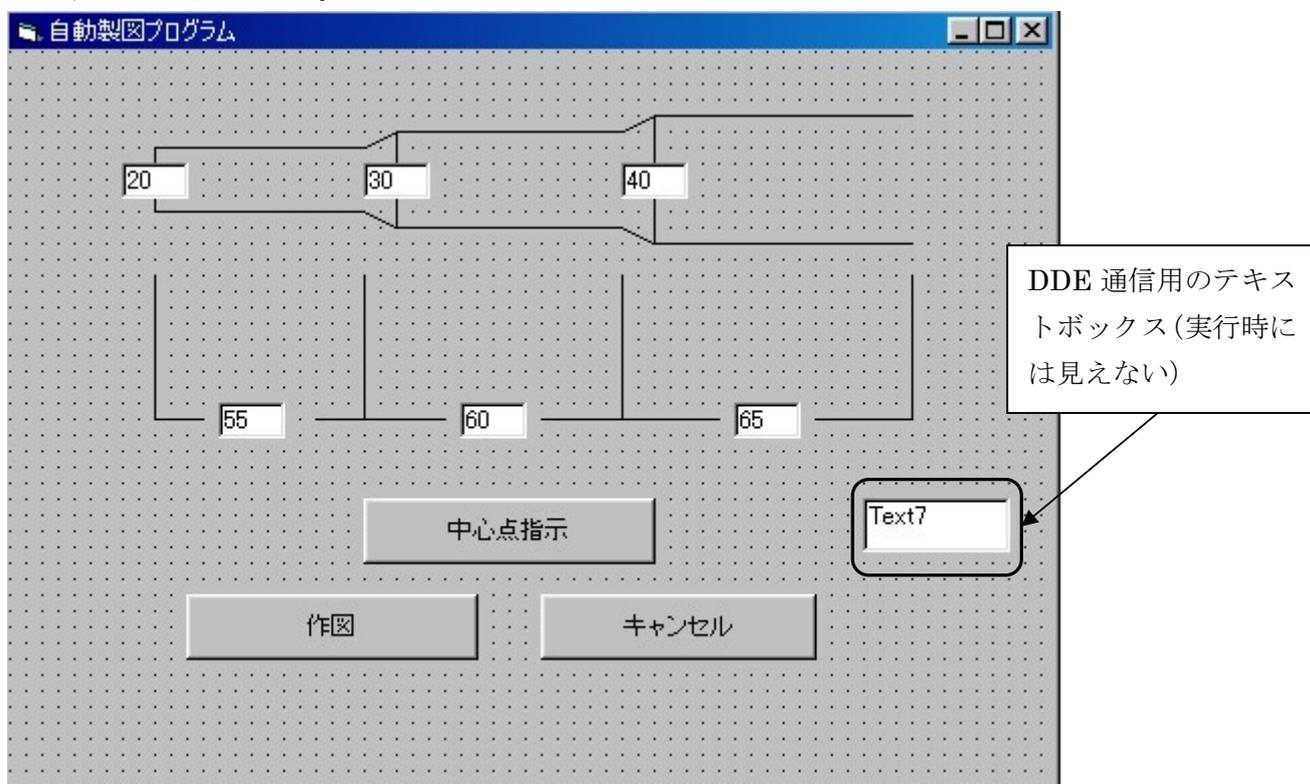


図 4-12. 『自動製図プログラム』ダイアログボックスの記述

以上の記述が実行されれば,Visual Basic から,AutoCADLT にコマンドを送れるようになる。ここまでの深い意味は考えなくても,決まった手順として扱えばよい。なお,実際に AutoCADLT にコマンドを送る文は,たとえば,以下のようなになる。

<AutoCADLT にコマンドを送る文の記述>

```
acadtext.LinkExecute コマンド文字列
```

コマンドボタンの記述

コマンドボタンをクリックして AutoCADLT にコマンド文字列を送るようには,エディットウィンドウ『HDRAW』で,『オブジェクト : Command\*』『プロージャ : Click』に切り替え,次のように記述する。

<コマンドボタン 1 の記述>

```
Private Sub Command1_Click ()  
    Dim cmd As String  
    Cmd = "ID" & Chr (13)  
    Acadtext. LinkExecute cmd  
End Sub
```

なお,コマンド『Cmd』は,ID コマンドで点の座標を求めるものである。『Chr (13)』は,『エンターキー』を押したのと同じ意味である。つまり,コマンドボタン 1 のクリックという動作によって AutoCADLT で『ID』とキーボードから入力して『エンターキー』を押すという手順が送信されることになる。

座標計算は,相対座標になるように計算している。また,座標値の前には相対座標を示す『@』をつけている。

数値を文字列に変換するには Format 関数を使っている。これは,Str 関数では文字列の頭にスペースが 1 文字入り,扱いにくくなるためである。

『Debug.Print cmd』は,開発時にコマンド文字列の内容を確認するために入れた。

このプログラムでは,簡略化のためエラー処理や入力データのチェックなどは省いてある。実務に使う場合にはこれらの点について配慮する必要がある。

<コマンドボタン 2 の記述>

```
Private Sub Command2_Click()
```

```
    Dim x1 As Double, x2 As Double
```

```
    Dim x3 As Double, y1 As Double
```

```
    Dim y2 As Double, y3 As Double
```

```
    Dim p(13) As String
```

```
    Dim cmd As String
```

```
    x1 = Val(Text1.Text)
```

```
    x2 = Val(Text2.Text)
```

```
    x3 = Val(Text3.Text)
```

```
    y1 = Val(Text4.Text)
```

```
    y2 = Val(Text5.Text)
```

```
    y3 = Val(Text6.Text)
```

入力した数値を文字で置換する設定

```
    p(1) = "@0," & Format(y1 / 2) & " "
```

```
    p(2) = "@ " & Format(x1) & "," & Format(0) & " "
```

```
    p(3) = "@ " & Format(((y2 - y1) / 2 * 5) & "," & Format(y2 / 2 - y1 / 2) & " "
```

```
    p(4) = "@ " & Format(x2) & "," & Format(0) & " "
```

```
    p(5) = "@ " & Format((y3 - y2) / 2 * 5) & "," & Format(y3 / 2 - y2 / 2) & " "
```

```
    p(6) = "@ " & Format(x3) & "," & Format(0) & " "
```

```
    p(7) = "@ " & Format(0) & "," & Format(-y3) & " "
```

```
    p(8) = "@ " & Format(-(x3) & "," & Format(0) & " "
```

```
    p(9) = "@ " & Format(-((y3 - y2) / 2 * 5)) & "," & Format(y3 / 2 - y2 / 2) & " "
```

```
    p(10) = "@ " & Format(-(x2) & "," & Format(0) & " "
```

```
    p(11) = "@ " & Format(-((y2 - y1) / 2 * 5)) & "," & Format(y2 / 2 - y1 / 2) & " "
```

```
    p(12) = "@ " & Format(-x1) & "," & Format(0) & " "
```

```
    p(13) = "@0," & Format(y1) & " ""
```

座標の設定

## 5章 電子計算書プログラムの構築

### 5-1 電子計算書プログラムの概要

電子計算書では、エクセルの関数使用と自動計算により設計計算を行い、その結果をワードで作成された成果品に『リンク貼り付け』機能を用いて取り込んでいる。『リンク貼り付け』機能はエクセルの計算結果からワードの内容を自動的に更新するのに用いている。図3及び図4に、エクセルの設計計算システムとワードの成果品の一部を示す。図3の中で、赤色の数値は設計者が設定すべき値であることを、緑色の数値は設計条件として与えられる値であることを、青色の数値は特に『注意』すべき値であることをそれぞれ示している。

	A	B	C	D	E	F	G	H	I	J	K
1	単純非合成桁橋										
2	設計条件										
3	1.形式										
4	2.設計活荷重										
5	3.床版										
6	4.支間	33000	33 m	片方		片方		支間/6			
7	4.橋長	33500	33.5 m	増やした厚さ		溝の幅		5.5 m			
8	5.幅員	7600	7.6 m		0.1 m		0.15 m		5500		
9	6.主桁高						0.25		550		
10	7.主桁本数		4本				250				
11	8.横桁本数		1本								
12	9.主桁間隔	L2	2.3 m								
13	10.舗装	as	車道	70 mm		0.07 m					

図 5-1. エクセル上での設計計算

例えば、エクセルにおいて主桁本数として4本と入力すると、リンク貼り付けが自動的に行われワードでも主桁4本と表示される。このようにエクセル上で設計計算をすることで、自動的にワード形式の成果品が完成する。

1. 設計条件	
形 式	主桁並列形式・単純非合成鋼プレートガーダー橋
設計活荷重	B活荷重
床 版	鉄筋コンクリート床版
橋の支間	33 m
橋の幅員	7.6 m
主桁本数	4本 ← 自動的に更新される
横桁本数	1本
横断勾配	2% 放物線
許容応力度	鉄 筋 引張応力度 $\sigma_{sa} = 1200 \text{ kg f/cm}^2$ (SD295)
	コンクリート 圧縮応力度 $\sigma_{ca} = \sigma_{ck}/3$
	$= 240/3 = 80 \text{ kg f/cm}^2$
	( $\sigma_{ck} = 240 \text{ kg f/cm}^2$ ; 設計基準強度)
1日当りの大型車の交通計画量 (台/日)	1000以上 2000未満

図 5-2. ワードによる成果品の自動作成

次に、ワード側から入力内容を修正する方法を述べる。これには、ワード側で編集すべき項目に対して『リンクの編集』を施すことで実現される。例えば、電子計算書で幅員が7.6mとなっているのを6.8mと訂正したい場合、『橋の幅員; 7.6m』の7.6という赤色部分を選択し、右クリックして『リンクの編集』という項目を選択する。するとエクセル画面が開き、リンクされた場所へカーソルを合わせた状態で表示されるので、そのセルを修正すればワードの内容は自動的に更新される。他に修正箇所がある場合も同様の手順で修正することができる。

## 5-2 電子計算書の使い方

最初にエクセルを起動させ、次にワードを起動させる。そしてワードのほうで数値を設定する。設定すべきところは赤く色がつけられている。

例えば、『33m』を『32m』と修正したい場合は、赤い数値を **33m** 選択し、右クリックすると **図 5-3** のようになる。その項目の中から『編集 (E) リンク』を選択する。



図 5-3. ワードからの編集

	A	B	C	D
1	単純非合成I桁橋			
2	設計条件			
3	1.形式			
4	2.設計活荷重			
5	3.床版			
6	4.支間	33000	33 m	
7	4.橋長	33500	33.5 m	
8	5.幅員	7600	7.6 m	

図 5-4. エクセル編集

すると、エクセル画面 ( **図 5-5** ) に切り替わりワードの『リンク貼りつけ』で指定した数値にカーソルが自動的に指定される。

そのセル **33** を『32』 **32** に入力し直せば、自動的にワードのほうも更新される。( **図 5-6 参照** )

他の場所も修正する場合は、同様の操作で修正する。

**1. 設計条件**

形式 ; 主桁並列形式・単純非合成鋼プレートガーダー橋

設計活荷重 ; B活荷重

床版 ; 鉄筋コンクリート床版

橋の支間 ; **32m** ← 自動更新された後

橋の幅員 ; 7.6m

主桁本数 ; 4本

横桁本数 ; 1本

横断勾配 ; 2% 放物線

許容応力度 ; 鉄筋 引張応力度  $\sigma_{sa} = 1200 \text{ kg f/cm}^2$  (SD 295)

コンクリート 圧縮応力度  $\sigma_{ca} = \sigma_{ck} / 3$

$= 240 / 3 = 80 \text{ kg f/cm}^2$

( $\sigma_{ck} = 240 \text{ kg f/cm}^2$  ; 設計基準強度)

1日当りの大型車の交通計画量 (台/日) ; 1000以上 2000未満

図 5-5. 編集終了後

## 6章 エクセルについて

マイクロソフト社の **Microsoft Excel**(以下エクセルと称す)とは、汎用の表計算ソフトである。ここでは、プログラムに使った関数を紹介する。

**IF** 文とは条件によって使う数式を分岐させて計算を行わせる関数である。では次に使い方を紹介する。

### 6-1 IF 文の使いかた

指定された条件が **TRUE** (真) のとき『**真の場合**』を返し、**FALSE** (偽) のとき『**偽の場合**』を返す。**IF** 関数を使用して値または数式が条件を満たしているかどうかをテストできる。

#### 書式

##### **IF(論理式, 真の場合, 偽の場合)**

**論理式** : 真または偽のどちらかに評価できる値または式を指定する。たとえば、**A10=100** は論理式である。この論理式は、セル **A10** の値が **100** の場合に **TRUE** となり、それ以外の場合に **FALSE** となる。引数には、どの比較演算子（後述参照）でも使用できる。

**真の場合** : 論理式が **TRUE** の場合に返す値を指定する。たとえば、この引数が“予算内”という文字列の場合、論理式が **TRUE** となったときに“予算内”という文字列が表示される。論理式が **TRUE** で『**真の場合**』が空白のときは、ゼロが返される。“**TRUE**”という語を表示するには、引数に論理値 **TRUE** を指定する。『**真の場合**』に、別の数式を指定することもできる。

**偽の場合** : 論理式が **FALSE** の場合に返す値を指定する。たとえば、この引数が“予算外”という文字列の場合、論理式が **FALSE** となったときに“予算外”という文字列が表示される。論理式が **FALSE** で『**偽の場合**』を省略（『**真の場合**』の後のカンマ以降を省略）すると、論理値 **FALSE** が返される。論理式が **FALSE** で『**偽の場合**』が空白（『**真の場合**』の後にカンマと閉じるかっこが続く）であると、ゼロが返される。『**偽の場合**』に、別の数式を指定することもできる。

#### 解説

- 最大 7 つまでの **IF** 関数を『**真の場合**』、『**偽の場合**』としてネストすることにより、より複雑なテストを行うことができる。（詳細は、最後の使用例を参照）
- 『**真の場合**』、『**偽の場合**』が選択されるとき、それらの計算結果の値が返される。
- いずれかの引数が配列である場合、**IF** 関数が実行されるとき、それぞれの配列要素はすべて選択される。
- **Excel** には、条件を基にデータを解析できる他の関数がある。たとえば、文字列またはセルの範囲内にある数値の個数をカウントするには、**COUNTIF** ワークシート関数を使用する。文字列または範囲内の数値に基づいて合計を計算するには、**SUMIF** ワークシート関数を使用する。

## 使用例

予算シートのセル A10 には、現在の予算を計算する式が入力されている。IF 文の例 1 では、A10 の式の結果が 100 以下の場合には“予算内”と表示され、それ以外の場合には“予算外”と表示される。

IF (A10<=100, “予算内”, “予算外”)

	A	B	C	D	E	F	G	H
1	予算内							
2	158							
3								

図 6-1. IF 文の例 1

IF 文の例 2 では、セル A10 に 100 という値が入力されている場合、『論理式』が TRUE となり、セル範囲 B5:B15 の合計が計算される。これ以外の場合には『論理式』が FALSE となり、IF 関数が記述されているセルには、空白文字列 (“”) が返される。

IF (A10=100, SUM (B5:B15), “”)

	A	B	C	D	E	F
1	予算内					
2	158					
3						
4						
5		10				
6		11				
7		12				
8		13				
9		14				
10	100	15				
11		16				
12		16				
13		16				
14		17				
15		18				

図 6-2. IF 文の例 2

IF 文の例 3 は、1 月から 3 月までの実際の支出と予想支出を記録したワークシートがある。セル範囲 B2:B4 には、実際の支出 1500000、500000、500000 が入力されていて、セル範囲 C2:C4 には、予想支出 900000、900000、925000 が入力される。

次のように IF 関数を利用すると、それぞれの月で予算が超過していないかどうかチェックして、その結果によって異なるメッセージを表示することができる。

	A	B	C	D
1				
2		1500000	900000	予算超過
3		500000	900000	OK
4		500000	925000	

図 6-3. IF 文の例 3

IF (B2>C2, “予算超過”, “OK”) = “予算超過”

IF (B3>C3, “予算超過”, “OK”) = “OK”

次の例では、“平均点”という名前で参照される点数に対して、A から E までの評価を与える場合を考える。

“平均点”の値	与える評価
90 点以上	A
80 点以上 90 点未満	B
70 点以上 80 点未満	C
60 点以上 70 点未満	D
60 点未満	F

図 6-4. 評価基準の表

	A	B	C	D
1	クラス	平均点	評価	
2	3-A	67.2	D	
3	3-B	20.1	F	
4	3-C	90.5	A	

図 6-5. 計算結果

このような処理を行うには、次のように IF 関数をネストさせて実行する。

`IF(平均点>89, "A", IF(平均点>79, "B", IF(平均点>69, "C", IF(平均点>59, "D", "F"))))`

この例で、2 番目の IF 関数は最初の IF 関数の『偽の場合』になっており、3 番目の IF 関数は 2 番目の IF 関数の『偽の場合』になっている。たとえば、最初の『論理式』（平均点 > 89）が TRUE のとき評価は “A” となり、FALSE のとき 2 番目の IF 関数が計算される。

『数値』を四捨五入して指定された『桁数』にする。

## 書式

### ROUND(数値, 桁数)

**数値** : 四捨五入の対象となる数値を指定する。

**桁数** : **数値** を四捨五入した結果の桁数を指定する。

- 『桁数』に正の数を指定すると、『数値』は小数点の右側（小数点以下）で四捨五入され、小数点以下の桁数が『桁数』に等しくなる。
- 『桁数』に 0 を指定すると、『数値』は最も近い整数として四捨五入される。
- 『桁数』に負の数を指定すると、『数値』は小数点の左側（整数部分）で四捨五入される。

## 使用例

ROUND (2.15, 1) = 2.2

ROUND (2.149, 1) = 2.1

ROUND (-1.475, 2) = -1.48

ROUND (21.5, -1) = 20

	A	B	C
1	値	ROUNDの結果	
2	2.15	2.2	
3	2.149	2.1	
4	-1.475	-1.48	
5	21.5	20	

図 6-6. ROUND

演算子は、数式の要素に対して実行する計算の種類を指定する。演算子には、算術演算子、比較演算子、文字列演算子、参照演算子の 4 種類がある。

### 算術演算子

次に挙げる算術演算子は、加算、減算、乗算、除算、べき算などの基本的な計算を実行し、数値を組み合わせて、計算結果として数値を返す。

算術演算子	内容	例
+ (プラス記号)	加算	3+3
- (マイナス記号)	減算 負の数	3-1 -1
* (アスタリスク)	乗算	3*3
/ (スラッシュ)	除算	3/3
% (パーセント記号)	パーセンテージ	20%
^ (キャレット)	べき算	3^2 (3*3 と同じ)

**比較演算子** 次に挙げる比較演算子は、2 つの値を比較し、結果として TRUE または FALSE の論理値を返す。

比較演算子	内容	例
= (等号)	左辺と右辺が等しい	A1=B1
> (~より大きい)	左辺が右辺よりも大きい	A1>B1
< (~より小さい)	左辺が右辺よりも小さい	A1<B1
>= (~以上)	左辺が右辺以上である	A1>=B1
<= (~以下)	左辺が右辺以下である	A1<=B1
<> (不等号)	左辺と右辺が等しくない	A1<>B1

**文字列演算子** 文字列演算子 "&" は、複数の文字列を組み合わせて、1 つの文字列の値に結合する。

文字列演算子	内容	例
--------	----	---

& (アンパサンド) 2 つの文字列を結合、または連結 "North" & "wind" は  
して、1 つの連続する文字列の値 "Northwind" となる。  
を作成。

**参照演算子** 次に挙げる参照演算子は、計算のためにセル範囲を結合する。

参照演算子	内容	例
:(コロン)	セル範囲の参照演算子です。2 つのセル参照を含め、その間に含まれるすべてのセルによって構成される 1 つの参照を作成。	B5:B15
, (カンマ)	複数選択の参照演算子です。複数の参照を 1 つの参照に結合する。	SUM (B5:B15, D5:D15)

## 7章. 考察

自動製図については、簡単な図の作成に必要な Visual Basic6 で Auto CAD LT を動かすためのコードは記述できたが、より複雑な図を作成するコードを記述するまでには至らなかった。しかし、5章で

示した電子計算書からのデータ自動入力を実現すれば,基本的には自動製図は可能である。

電子計算書の作成においては,設計計算の半自動化のためいくつかの制約条件を課した。例えば,現場継ぎ手の設計においてボルトの配置を標準的なものに限定したことである。

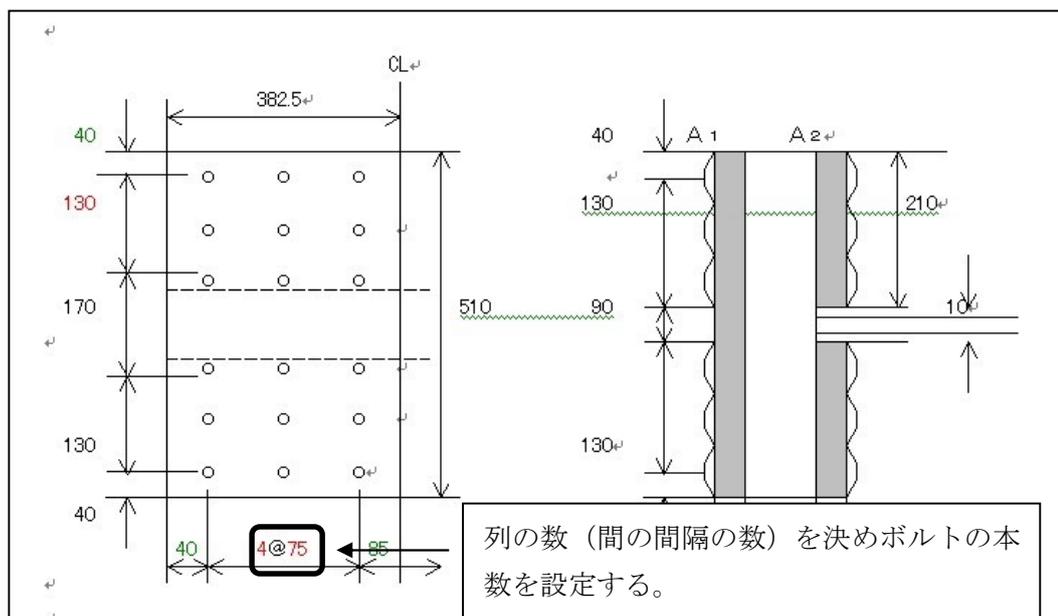


図 7-1. 現場継ぎ手 (上フランジのボルトの配置)

ボルトの本数の増減は一行増やすか減らすかで調節することで対応している。ここでのエクセルデータでは縦方向,横方向の座標を設定して計算処理している。ここでは上フランジのみ記したが他の下フランジや,現場継ぎ手の応力度照査のところも同様の方法で計算処理した。その他の設計条件としては皆川先生の,HP の設計製図の項目 (参考サイトにアドレスを記載) を参照。

その他に,電子計算書についての注意事項として研究資料にその成果品『設計製図 2.2 改完成 p』を,保存するときに Windows のデスクトップ上に『新しいフォルダ』という名前のフォルダを作り,そこに『設計製図 2.2 改完成 p』を保存しなければならない。なぜなら他の場所に保存すると,リンク貼り付けを指定した場所と異なってしまうのでリンク貼り付けの場所を設定し直さなければならないという欠点が生じるためである。以上が残りの欠点である。

しかし,本システムにより,設計及び成果品作成に要する時間は著しく短縮された。

## 8章. 終わりに

表計算ソフトにおける設計計算の結果を反映させて半自動的に電子設計書を作成するシステムの構築はほぼ完成したものの,それを用いてさらに半自動製図をするシステムについては基本的な動作

確認にとどまった。これについては今後の課題としたい。

#### 謝辞

本研究を進めるにあたって、皆川勝助教授、佐藤安雄技師より多くの助言をいただきました。ここに謹んで厚く御礼申し上げます。

参考資料・参考サイト

参考資料

- 1) 谷尻かおり：はじめての Visual Basic6,技術評論社, 1999.3.1.
- 2) 鈴木裕二,伊藤美樹：CAD 徹底解説シリーズ試せる Auto CADLT 徹底解説 LT97/98,建築知識,1992.2.10.
- 3) ノマド・ワークス：最新パソコン基本用語辞典,新星出版社,1998.12.15.
- 4) 社会法人 日本道路協会：道路橋示方書・同解説 I 共通編 II 鋼橋編,1996.12.19.

参考サイト

- 1) Add 設計 ホームページ (<http://www.adds.co.jp/>)
- 2) 皆川勝助教授 ホームページ (<http://cipc04.civil.musashi-tech.ac.jp/ci-j.html?>)

## 付録 1

# 卒業論文・中間発表概要集

## 第一回中間発表概要

第一回中間発表プレゼンテーション

## 第二回中間発表概要

第二回中間発表プレゼンテーション

## 第三回中間発表概要

第三回中間発表プレゼンテーション

## 第四回中間発表概要

第四回中間発表プレゼンテーション

## 第五回中間発表概要

第五回中間発表プレゼンテーション

## 第六回中間発表概要

第六回中間発表プレゼンテーション

## 最終発表概要

最終発表プレゼンテーション