

ISSN 0386—5878

土木研究所資料第3071号

001

土木研究所資料

旅行速度集計解析プログラム使用の手引

平成4年1月

建設省土木研究所
道路部新交通研究室

Copyright © (1992) by P. W. R. I.

All rights reserved. No part of this book may be reproduced by any means, nor transmitted, nor translated into a machine language without the written permission of the Director General of P. W. R. I.

この報告書は、土木研究所長の承認を得て刊行したものである。したがって、本報告書の全部又は一部の転載、複製は、土木研究所長の承認を得ずしてこれを行ってはならない。

旅行速度集計解析プログラム 使用の手引

新交通研究室 室 長 山田晴利

研究員 河野辰男

吉川 修

要 旨

道路の交通特性を把握するためには、旅行速度調査を常時体系的に実施する必要がある。また当研究室では、旅行速度調査の効率的な運用を図るため旅行速度測定装置の開発を進めてきている。本報告書は旅行速度測定装置を使用して旅行速度常時観測を実施するための、集計プログラムの使用の手引である。

キーワード：旅行速度調査，機械観測，渋滞調査，プログラム，使用説明書

はじめに

時々刻々と変化する交通状況を調査しその路線の交通特性を把握する上で、交通量の他に旅行速度や渋滞長等も重要な情報であり、かつ経時変化を捉えることが望まれる。特に、都市部の路線では渋滞対策を効果的なものとする観点からもこれらの状況をモニタリングする意義は大きい。このため、旅行速度調査を常時体系的に把握し、道路の整備効果や交通状況を把握する必要がある。

現在行われている旅行速度の調査方法の代替案として、新交通研究室では旅行速度測定装置を開発した。この旅行速度測定装置は、走行速度を自動計測し、容易にかつ高精度で数多くのデータを収集できるものである。

旅行速度測定装置を使用して旅行速度の常時観測を実施するに当たり、旅行速度測定装置のデータをパソコンによって集計するプログラムを作成した。

本報告書はこの集計プログラムの使用の手引とそのプログラムリストである。

旅行速度測定装置は各地方建設局、北海道開発局、沖縄事務局で1台以上は購入済（平成4年1月現在）であり、集計プログラムも各部所へ配布されている。

全国10の地方建設局、北海道開発局、沖縄事務局においては、旅行速度調査を平成3年10月より1年間実施している。このデータの収集の後、調査手法、プログラムについて再度検討を行い、旅行速度観測のための指針を取りまとめる予定である。

平成4年1月

建設省土木研究所
道路部新交通研究室

旅行速度集計解析プログラム

使用の手引

目 次

はじめに

1. 概要	1
1. 1 プログラムの概要	1
1. 2 解析作業の手順	2
2. プログラムの基本操作	5
2. 1 プログラム起動と終了	5
2. 2 データ入出力	7
3. プログラムの使用方法	8
3. 1 路線データの作成	8
3. 2 走行状態の解析	16
3. 3 渋滞情報の集計	23

付録

1. 道路データ、路線データの記録様式	27
2. 調査データの記録様式	32
3. 漢字入力の方法	34
4. スクリーンエディタの使い方	35
5. プログラムリスト	38
6. 旅行速度常時観測調査 実施要領	86
7. 旅行速度測定装置の概要	97

1. 概要

1. 1 プログラムの概要

本プログラムパッケージは、建設省土木研究所にて開発した「旅行速度測定装置」を用いて行なう旅行速度常時観測調査結果を解析・集計するためのものである。

1) 機器構成

プログラムが動くために必要な機器は、NEC製PC9801シリーズ又はエプソンのパーソナルコンピュータで、最小機器構成は図1-1に示すとおりである。

なお、640kbのメモリーとMS-DOSシステムの基本ソフトが必要である。

また、本プログラムパッケージは、フロッピーベースで作業することを前提としているため、フロッピーディスクを挿入する装置番号が固定されているので注意する必要がある。

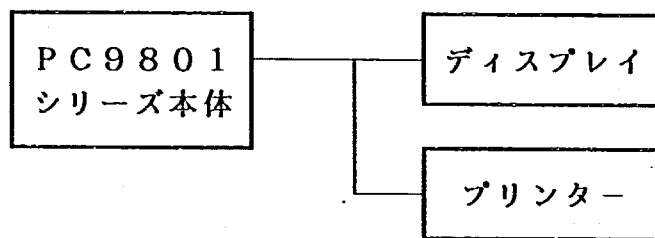


図1-1 必要最小機器構成

また、調査解析に先立ち以下の3種類のフロッピーディスクを準備する。

- ① プログラムディスク（土木研究所より送付のもの） 1枚
- ② 集計用フロッピーディスク 1枚
- ③ 調査用フロッピーディスク 必要枚数

なお、②、③のフロッピーディスクは、MS-DOSにてフォーマットをしておく必要がある。

2) プログラム対象利用者

本プログラムを使用する者は、パーソナルコンピュータ及びMS-DOSの基本的な操作ができることを前提としている。そのため、機器の操作、フロッピーディスクの扱い、MS-DOSの基本コマンドによる操作などについては既に知っているものとし、本使用の手引では解説しない。

1. 2 解析作業の手順

プログラムパッケージの構成及び解析作業の流れを示すと図1-2のとおりである。

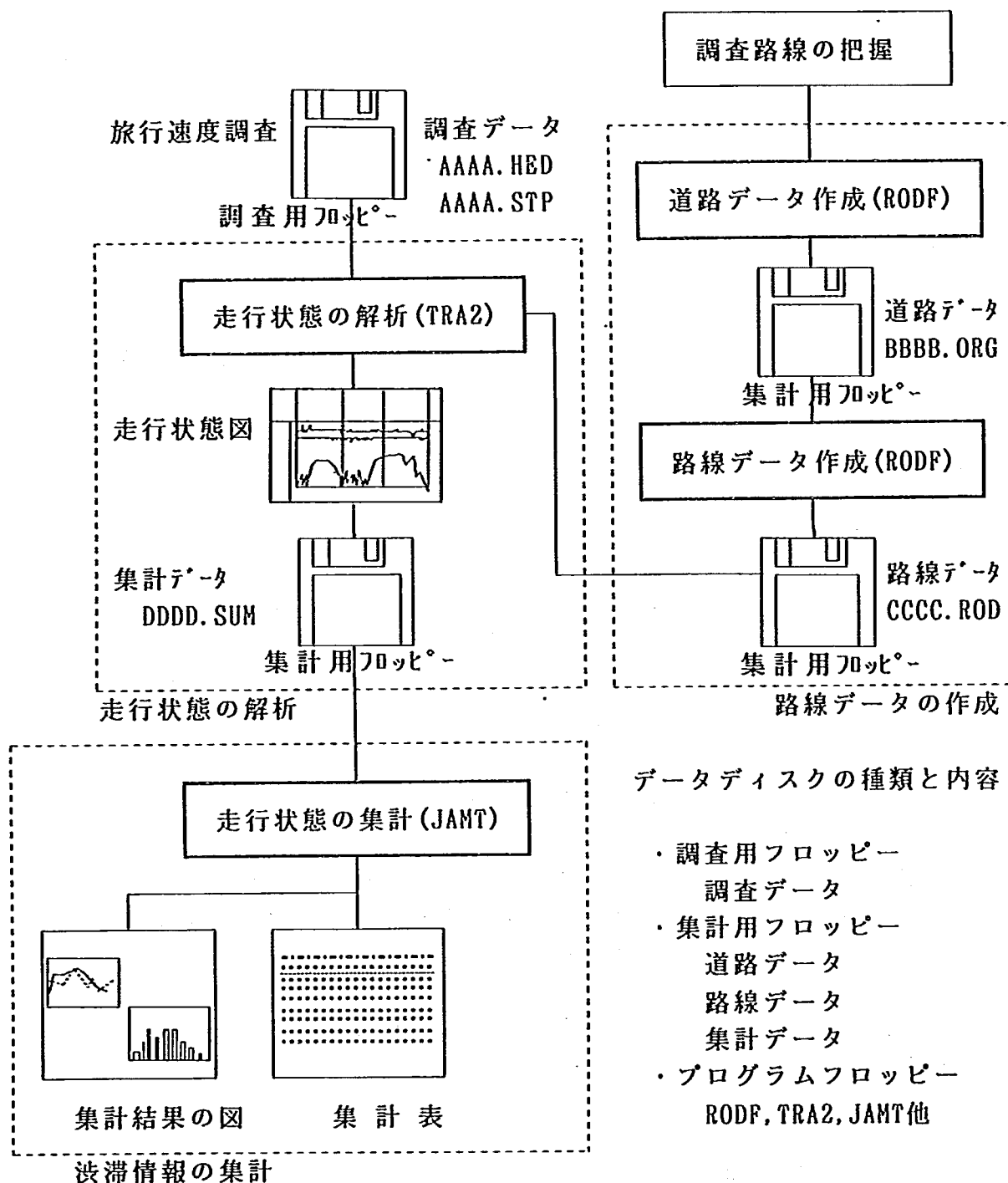


図1-2 旅行速度調査（常時観測）解析手順

なお、本手引きでは、データファイルについて単に「データ」と呼び、このデータなどの入っているディスクを「フロッピー」または「フロッピーディスク」と呼ぶことにする。

旅行速度の常時観測データの解析を行うための手順を以下に示す。

1) 調査対象区間の選定

調査対象区間は、複数の交通センサス区間に対応した約10km程度の区間とし、渋滞が問題となっている区間とする。

2) 道路データの作成

調査対象区間の道路情報をまとめた道路データを作成する。道路データは、道路種別、車線数や、センサス区間内の交差点をキロポストの昇順に並べたものであり、これを基に、実際の解析に使用する路線データが作成される。

この道路データの作成作業は、メインメニューより「路線データの作成」を選択して行うことができるが、作業を開始する前にデータの記録例などを参考に必要なデータを収集しておく必要がある。

3) 路線データの作成

道路データを基に、解析に使用する路線データを作成する。路線データは、道路データの情報に従って、調査を実施する方向（上り下り）を考慮して作成されるものであり、道路データと同様のプログラムにより簡単に作成できる。

なお、道路データ及び路線データの作成作業は、解析に先立ち1回実施すればよい。

4) 実査（詳細は、「旅行速度常時観測調査 実施要領」を参照のこと）

旅行速度調査の開始にあたり旅行速度測定装置の以下の設定を確認する。

- ① ステップ数は、「5」とする。
- ② 日付、時刻を合わせる。
- ③ コメント番号に路線番号を設定する。
- ④ 前進・後退が正しいかチェックする。

試験車の走行は、他の車両の流れに沿って走行するものとする。また、調査区間の起点・終点におけるスタート、ストップスイッチの操作を確実に行う。

走行調査時のマーカ入力は、表1-1に示すとおりとする。このマーカ情報は、渋滞原因として利用するので、当該箇所の通過時に遅滞なく入力するようにする。

実査にあたっては、「実施要領」に示す調査記録簿に、開始・終了日時、天候、異常事象の発生などを記録しておくと共に、調査データファイル名（付録参考）との確認をしておく。

表 1 - 1 マーカ種別

マ ー カ	原 因
1 交通事故	交通事故地点
2 道路工事	道路工事地点
3 幅員狭少	幅員狭少、その他

5) 走行状態の解析

走行状態解析プログラムを用い、走行状態図及び集計データを作成する。この作業は、基本的には走行調査データを指定すると自動的に行われるが、「天候関係」及び「時間交通量」については担当者が入力しなければならない。特に時間交通量については、交通量常時観測地点の調査日時に該当する交通量を入力しなければならない。

集計データは、解析を行うと順次結合される。なお、同一の日時、方向の調査結果を解析した場合、集計データには、後から解析した結果が残される。

6) 走行状態の集計

集計データより時間帯別平均旅行速度図をはじめとする各種集計を行うものであり、メインメニューより集計プログラムを起動することによって自動的に行われる。

2. プログラムの基本操作

2. 1 プログラム起動と終了

1) データ

本プログラムパッケージでは、多くのデータを使用するため、データの種類毎に以下のようなデータ名の規則を設けてある。

データ名は、「xxxxxxx.yyy」という形をしており、ピリオドの前半部分が本来のデータ名であり、ユーザが自由につけることができる。但し、調査データについては、旅行速度測定装置により、測定開始月・日・時・分（10分単位）と積算スタート回数によりデータ名がつけられるので、そのまま利用することが望ましい。また、プログラムによる処理を実行する場合に入力を要求されるデータ名は、この前半部分であり、後半部分は、入力しないものとする。

データ名の後半部分（拡張子）については、既定値が以下のとおり定められており、変更することができない。

表 2-1 ファイル名称一覧表

拡張子	ファイルの内容	フロッピーディスクの種別
. HED	調査データのヘッダー情報	調査用フロッピー
. STP	調査データのステップ情報	
. ORG	路線別の道路データ	集計用フロッピー
. ROD	調査経路別の路線データ	
. SUM	集計データ	

なお、フロッピーディスクをデータディスクとして使用するためには、必ず初期化を行わなければならない。また、古いディスクを今後データディスクとして用いる場合も、この初期化を行ったほうがよい。

プログラムディスクを含むフロッピーディスクは、以下の装置に挿入しなければならない。

プログラムディスク	装置 A
調査用フロッピー	装置 A
集計用フロッピー	装置 B

なお、走行調査データの入っている調査用フロッピーは、プログラムの指示に従って、プログラムディスクと交換するようにすること。

2) プログラムの起動

プログラムパッケージの起動は、以下の手順で行う。

- ① プログラムディスクをディスク装置 A に挿入する。
- ② 電源を ON にする。

以上の操作で、MS-DOS が自動的に立ち上がり、メインメニューが出るので、処理したい項目をメニューから選択することによって希望するプログラムを起動できる。

3) プログラムの終了

プログラムの終了は、メインメニューの [終了] を選択することで MS-DOS の入力待ち状態となるので、ディスクを抜いた後、電源を OFF にする。

4) メインメニュー

本プログラムパッケージの処理の選択は、メインメニューによって行う。

メインメニューは、図 2-1 に示すとおりである。

メニューの選択は、↑↓キーで希望するメニューに合わせ、リターンを押すことで行う。

図 2-1 メインメニュー

常時観測調査プログラム	メインメニュー (コマンド選択) 1/1	Menu v2.20
F1 路線データの作成		日付: 1991-10-22 時間: 09:46 MS-DOS: Ver. 4.00
F2 路線データの印字		
F3 走行状態の解析		
F4 渋滞情報の集計		
F5 作業の終了		
F6		
F7		
F8		
F9		
B>RODF		
路線データを作成・修正します。		
矢印で項目を選択し、リターンキーを押して下さい		

2. 2 データ入出力

1) データ入力の方法

キーボードからのデータ入力が必要な場合、画面に入力要求が表示される。この要求に従って利用者は、必要なデータをインプットする。データ入力は、基本的には「リターン」キーを押すことによって完了する。そのため、「リターン」キーを押す前ならば「BS」キーによって修正が可能であるが、「リターン」キーを押したあとは修正ができない。「リターン」キーを押した後に間違いを発見した場合は、強制的にプログラムを終了し、再度メインメニューによるプログラムの選択から行わなければならない。

なお、データ入力には、データの終わりを示すために、データを入力せずに「リターン」キーのみを押すだけのものもある。

2) フロッピーディスクのセット

処理は、全てフロッピーディスクベースで行えるように考慮されている。ユーザは、必要に応じてフロッピーディスクの交換、挿入確認などを行うものとする。なお、プログラムパッケージを装置Aに、集計用フロッピーを装置Bに入れることを標準としている。

また、処理の途中でフロッピーディスクを交換することは、プログラムによって要求された場合以外は行ってはならない。

3) プリンター出力

プリンターへの出力は、「路線データの印字」、「走行状態の解析」および「走行状態の集計」において行われる。

ページプリンタ（レーザプリンタ）を利用する場合には、用紙及びその方向は、「B4横」とするとよい。ドットプリンタを利用する場合は、15インチの連続紙を用いるものとする。なお、ドットプリンタの場合には、ハードコピーの関係で改ページが正しく行われない可能性があるため、ページプリンタの利用を推奨する。

3. プログラムの使用方法

メインメニューにより処理を選択すると、それぞれの処理プログラムによって画面に操作方法、データの入力要求などが表示される。この指示に従って対話形式で機器の操作やデータ入力を行えばよい。そのため、ここでは、特に入力に際して注意が必要な事項に限り説明を付けるものとする。

3. 1 路線データの作成

交差点名称とそのキロポスト及び交通センサスの区間情報をまとめたデータを作成しておき走行状態の解析で使用する。すなわち、解析に路線データを用いることにより走行調査データの距離補正を自動的に実施し、集計データをセンサス区間毎に作成するための基準として利用できる。

1) 概要

路線データの作成とは、路線別の「道路データ」の作成と、道路データをもとにした集計データ作成に利用される「路線データ」の作成とがある。このうち、実際に多くのデータ入力が必要なものは、道路データの作成だけであり、路線データは、センサス区間を指定するだけで自動的に作成される。

路線データ作成プログラムは、下図に示す流れを持つものである。最初に道路データの作成を行い、続いて路線データの作成を実行する。

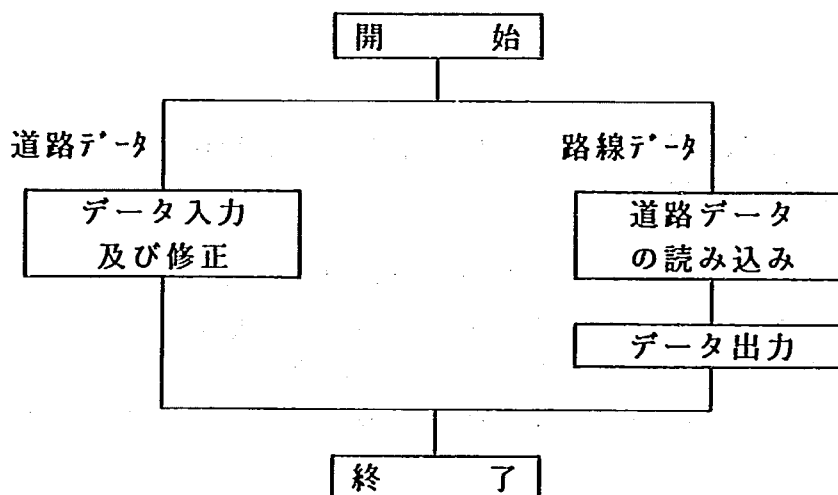


図3-1 路線データの作成手順

2) 路線データ作成メニュー

路線データ作成のメニューは、図3-2に示すとおり2種類があり、まず道路データを作成し、これをもとに路線データを作成するようにする。

* 作業選択 * 1. 道路データ入力・修正 2. 路線データの作成 3. 終了 番号を入力して下さい : ?

図3-2 路線データ作成メニュー画面表示例

3) 道路データの作成・修正

道路データは、路線名称「国道〇〇号」等の路線毎に起点から順にセンサス情報と交差点情報をまとめたデータである。

路線番号は、路線の名称の「国道〇〇号」等の〇〇に相当する番号で最大4桁の英数字である。

道路データファイル名は、ファイルの名称であり、*****.ORGの*****に相当する最大8桁の英数字である。

道路データの作成にあたっては、道路データファイル名が表示され、選択にあたっては、路線番号で入力するため、路線番号と道路データファイル名は同じ英数字を入力するのが望ましい。(1対1で対応していれば違っていても問題はない)例えば、国道14号の場合、路線番号は 14と入力し、道路データファイル名も 14のようにすると判別しやすくなる。なお、拡張子の「ORG」は、自動的に付けられるので指定する必要はない。

道路データの作成では、新規に作成する場合の他に、既に作成されている道路データの内容を修正する機能も持っている。データの 입력は、画面に表示される入力要求に従って行えばよい。また、データ入力終了した場合、あるいは現在の作業が終了した場合には、入力要求に対し、[リターン]キーのみ入力すると1つ前の入力状態になる。そのため、リターンのみの入力の繰り返しによって最初のメニューに戻ることができる。

なお、入力データのなかで、「キロポスト」については、昇順となっているかをチェックするが、他の入力データについてはプログラムにチェックする機能はないので、利用者の責任でチェックする。

道路データの作成手順は、以下のとおりである。

① 路線の指定

図3-2に示す路線データ作成メニューより「1」の道路データ入力・修正を選択すると、図3-3に示すように集計用フロッピーに存在する道路データファイル名の一覧表が表示され、路線番号の入力要求が出される。入力された路線番号と同じ道路データが集計用フロッピーに存在する場合には図3-4に示すように、道路データの修正とみなされ、センサス区間番号及び地点データ（キロポスト、地点名）の追加、修正、削除ができ、その他の場合は、新規に道路データを作成することになる。

* 道路データ入力・修正（道路ファイル選択） *

21.ORG 247.ORG 302.ORG

路線番号を入力して下さい： 21

図3-3 道路データ作成表示例（その1）

* 道路データの修正 [21.ORG] *

処理を選択して下さい(1:センサス区間修正 2:地点データ修正)

図3-4 道路データの作成表示例（その2）

新規作成の場合は、図3-5、6に示すように路線番号、道路データファイル名の入力、路線名称の入力を行う。

* 道路データ入力・修正（道路ファイル選択） *

21.ORG 247.ORG 302.ORG

路線番号を入力して下さい： 23

道路データファイル名を入力して下さい： 23

図3-5 道路データ作成表示例（その3）

* 道路データ作成 [23.ORG] *

路線名称を入力して下さい： 国道23号

図3-6 道路データ作成表示例（その4）

② センサス区間情報の入力

つづいて、図3-7に示すセンサス区間情報の入力を行う。

* 道路データ作成 ([23.ORG] 路線名称: 国道23号) *	
センサス調査区間	
都道府県番号を入力して下さい	: 23
市区町村番号を入力して下さい	: 111
センサス区間番号を入力して下さい	: 131
道路種別を入力して下さい	: 3
車線数を入力して下さい	: 3

図3-7 道路データ作成表示例(その5)

都道府県コードは2桁の数字、市町村コードは3桁の数字で入力する。また、センサス区間番号を4桁の数字で入力する。なお、先頭の「0」は、入力する必要はない。

「道路種別」は、センサスの調査要領に従った、以下のコードを入力する。

- 1: 高速自動車国道
- 2: 都市高速道路
- 3: 一般国道
- 4: 主要地方道(都道府県道)
- 5: 主要地方道(指定市市道)
- 6: 一般都道府県道

「車線数」は、当該センサス区間の平均的車線数(片側)を入力する。

③ 地点情報の入力

図3-8に示す入力要求に従って、主要信号交差点についてキロポスト順にキロポスト(km単位)と地点名称を入力する。

キロポストは、km単位で入力されたものが10m単位に四捨五入されてファイルに記録されるので、1m以下の細かい数値は入力する必要はない。また、バイパスなどキロポストが連続していない区間については、調査区間の延長と等しくなるように起点からの距離を入力するようにするとよい。

地点名称は、半角文字で22文字(漢字で11文字)以内で入力する。なお、地点名は省略してもよいが、できるだけ入力することが望ましい。

1つのセンサス区間(分割されている場合は、分割区間)内の地点数の最大は50地点とする。

キロポスト(Km単位)を入力して下さい : 359.90
地点名称を入力して下さい : 築地インター東

図3-8 道路データ作成表示例(その6)

センサス区間内の地点情報をすべて入力した場合、キロポストの入力に対し、[リターン]キーのみを入力すると、図3-9の表示がでるので、必要な番号を入力する。

センサス区間が分割されている場合は、分割区間が終了する毎にこの表示による選択を行う。

キロポスト(Km単位)を入力して下さい : 361.00
地点名称を入力して下さい : 寛政インター
キロポスト(Km単位)を入力して下さい :
センサス区間を分割しますか(1:分割区間入力 2:次のセンサス区間) :

図3-9 道路データ作成表示例(その7)

なお、地点情報の入力中に入力ミスをした場合には、そのまま入力を行き、全てのデータ入力終了した後、道路データの修正を行った方がよい。

漢字の入力方法、データの修正をエディタにて行うこともでき、その方法については、「付録」に述べてあるので、必要に応じて参照すること。

④ 道路データの修正

道路データの修正が選択された場合、センサス区間を指定する。指定されたセンサス区間に含まれる全地点が画面に表示されるので、追加、削除、地点名変更などを行う。

* 地点データ修正 (23.ORG センサス区間番号:0133) *

処理を選択して下さい(1:追加 2:削除 3:地点名変更) : 1

363.60	宝神交差点	363.75	庄内新川橋東交差点
364.30	庄内新川橋西交差点	365.00	藤前西交差点(信号無)
365.45	流通団地交差点	366.10	名古屋市・飛鳥村境

キロポストを入力して下さい :

図3-10 道路データ作成表示例(その8)

4) 路線データの作成

路線データは、走行状態の解析及び集計データの作成で使用するもので、走行調査を行うルートに従って、先に作成された複数の道路データを組み合わせて必要情報をまとめたものである。

路線データファイル名は、*****.ORGの*****に相当する最大8桁の英数字をである。

路線データのデータファイル名として調査経路が判別できるものが望ましい。

路線データは、作成のみで修正はできない。修正したい場合は、新たに作成しなければならない。

① 路線データファイル名の入力

路線データの作成を選択すると、集計用フロッピーに存在する路線データの一覧表が表示され、路線データファイル名の入力を要求される。入力したデータ名と同様のデータがフロッピーにある場合、新しく作成される路線データが上書きされるので注意すること。

* 路線データ作成 *

111.ROD 222.ROD

路線データファイル名を入力して下さい : 222

図3-11 路線データの作成表示例(その1)

② 路線の指定

つづいて、路線データを作成する基になる道路データを指定する。この場合、道路データファイル名の一覧表が表示されるが、入力するのは路線番号であり、指定された路線番号を持つデータが自動的に選択される。なお、同一の路線番号を持つ道路データが複数存在する場合の処理は保証されないので、1路線につき道路データは1つだけ作成するようにする。(図3-12の例では、道路データファイル名 23.ORGの中の路線番号が23となっていることを想定している。)

* 路線データ作成(道路ファイル選択) *

23.ORG 247.ORG 302.ORG

路線番号を入力して下さい : 23

図3-12 路線データの作成表示例(その2)

③ センサス区間の指定

路線番号を指定すると、該当する道路データに存在するセンサス区間番号の一覧表が表示されるので、センサス区間番号を入力する。センサス区間が分割されている場合でもセンサス区間番号の入力のみでよく、分割番号を入力する必要はない。

また、調査方向を番号で選択する。

* 路線データ作成 (222. ROD) ← [23. ORG] (センサス区間選択) * 0131 0132 0133 センサス区間番号を入力して下さい : 0133 調査方向 (1:上り 2:下り) : 2

図 3 - 1 3 路線データの作成表示例 (その 3)

ここまでで、1つのセンサス区間のデータが作成され、③に戻るので次のセンサス区間番号を入力する。また、他の道路データを続ける場合は、②に戻るまで単に [リターン] キーを押し、路線番号の入力から行う。この操作により、複数の路線を経由する路線データを作成することが出来る。なお、複数の路線の道路データを用いる場合、これら全ての道路データは、1つの集計用フロッピーに存在しなければならない。

5) 路線データの印字

路線データの内容をプリンターに出力する場合は、メインメニューより「路線データの印字」を選択し、画面に表示されるデータ名の中から印字したいものを選択することで行われる。

3. 2 走行状態の解析

走行調査データより、走行状態の解析に必要な情報を図表の形で表示するものである。また、解析結果を集計データとして出力する。

解析フローは、以下に示すとおりであるが、プリンターへの出力は、走行状態図の表示時に指定すると行われる。また、路線データを入力せずに走行状態を見ることができるようになっている。

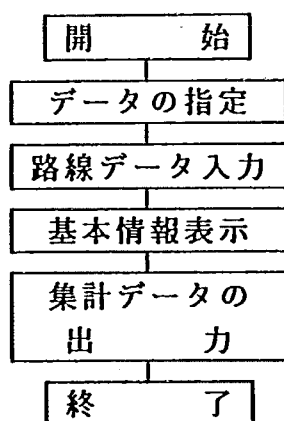


図3-14 走行状態の解析手順

1) 調査データの指定

走行状態の解析に先立ち、解析対象となる走行調査データを指定しなければならない。

本プログラムを起動した後、メッセージに従って調査データの入っているフロッピーディスクを装置Aに入っているプログラムディスクと交換すると、画面に調査フロッピーに入っている走行調査データの一覧表が表示されるので、矢印によりデータを指定する。

```
SELSCT FILE NEME (<SUB-DIR>, FILE , [ESC]QUIT , [RETURN] SELECT )
```

10210650.HED	10210711.HED	10210800.HED	10210811.HED	10210811.HED
10210900.HED	10210911.HED	10211000.HED	10211011.HED	10211100.HED
10211111.HED	10211210.HED			

図3-15 調査データ選択画面表示例

2) 路線データの入力

路線データを入力するか否かを聞いてくるので、一般には入力するようにする。なお、路線データを入力しない場合、集計データは作成されないので注意すること。

路線データを入力する場合は、画面の指示に従って路線データの入っている集計用フロッピーを装置Bに挿入する。路線データの inputs は、画面に集計用フロッピーに入っている路線データの一覧表が表示されるので、その中より路線データを指定すればよい。

```
SELSCT FILE NEME (<SUB-DIR>, FILE , [ESC]QUIT , [RETURN] SELECT )
```

```
111.ROD  222.ROD
```

図 3 - 1 6 路線データの選択例

3) 集計データ

集計データは、センサス区間毎の走行状態・渋滞状態をまとめたものでありその記録様式は、1つのセンサス区間毎に図 3 - 2 0 に示すとおりである。

集計データは、路線データと同じ集計用フロッピー上に作成され、その名称は、前半部分は、利用者の指定した名称となった上で、拡張子が「. SUM」となったデータとなる。なお、集計データを指定せずに単に「RETURN」のみ入力した場合には集計データは作成されない。

この集計データは、日常的な調査によって多くのデータを集め、後述する旅行速度調査結果の集計を行うことによって、路線別の渋滞状況の把握が可能となるものである。そのため、1つのファイルに多くの調査結果が結合され、走行状態の解析を行う毎に次第に大きくなってゆくものである。なお、同一日時の走行調査データを解析した場合には、最後に解析したものが集計データとして残されるので、まちがった解析を行った場合は、再度正しく解析すれば集計データは修正される。

現在の集計データ名

CHUBU.SUM

集計データ名を入力して下さい :

([RETURN] のみの場合は、集計データを作成しません)?

図 3 - 1 7 集計用データの選択例

4) 交通状況等の入力

「調査記録簿」に従って、天候、交通量などを入力する。入力、各項目に設定されたコード番号を入力する。また、交通量は、調査方向の時間交通量を入力する。なお、データのない場合には、「RETURN」のみ入力すればよい。入力を間違えた場合は、最後の確認で「N」を入力すると天候の入力に戻ることができる。

対象調査データは、10210711(10月21日 7時14分調査開始 コメント 3102) です

調査時の天候 (1:晴 2:曇 3:霧 4:小雨 5:雨 6:雪)? 1

調査時の視界 (1:良好 2:夕暮れ 3:暗がり)? 1

調査時の路面状況 (1:乾燥 2:湿潤 3:新雪 4:圧雪 5:凍結)? 1

当該方向の車種別時間交通量(台/時)を入力して下さい

全車種合計? 2385

大型車合計? 279

よろしいですか(Y/N)?

図3-18 交通状況の入力例

5) 解析終了

交通状況の入力が終了すると、走行状態の解析が自動的に実施されて走行状態図が作成され、ハードコピーをとるか否かを問い合わせてくる。ここでハードコピーをとるとして「Y」を入力した場合は、走行状態図のコピーと集計データの作成が行われる。ただし、最初に集計データの作成を指示していない場合は、集計データは作成されない。そのため、走行状態図を表示した時点で間違いを発見した場合などは、「N」を入力するとなにもせず終了させることができる。

その後、図3-19に示すようなメッセージが表示されるので、他の走行調査データを解析する場合は、「Y」を入力し、最初のデータ選択画面から繰り返す。解析を終了する場合は、「N」を入力し、画面の指示に従って、装置Aにプログラムディスクを挿入して[RETURN]を押すとプログラムは終了して、メインメニューに戻る。

次の調査データを解析しますか(y/n)? N

プログラムディスクを装置Aに挿入して下さい
確認したら [RETURN] を押して下さい

図3-19 終了画面例

項	目	桁	単 位
1	都道府県コード	2	
5	市町村コード	3	
10	調査単位区間番号 (センサスコード)	4	
15	分割区間番号	2	
20	路線番号	4	
25	道路種別	1	
30	区間延長	3	100m
35	調査方向	1	
40	調査方向車線数	1	
45	調査年月日	年 2 月 2 日 2 時 2	
50	天 候	1	
55	視 界	1	
60	路面状況	1	
65	起点通過時刻	時 2 分 2 秒 2	
70	終点通過時刻	時 2 分 2 秒 2	
75	時間交通量	全車 4 大型 4	台/時
80	走行時間	時間 2 分 2 秒 2	
85	停止回数	3	回
90	平均放行速度	4	0.1 km/h
95	平均加速度	4	0.01 m/s ²

項	目	桁	単 位
70	平均加速度	4	0.01 m/s ³
75	停止時間	4	秒
80	総渋滞長	ランク 1 4 ランク 2 4	10m
85	総渋滞区間通過時間	ランク 1 4 ランク 2 4	秒
90	渋滞区間内放行速度	4	0.1 km/h
95	最大渋滞発生位置	3	100m
100	渋滞長	3	100m
105	第二渋滞発生位置	3	100m
110	渋滞長	3	100m
115	第三渋滞発生位置	3	100m
120	渋滞長	3	100m

図 3 - 2 0 集計データの記録様式

6) 走行状態解析結果の見方

画面上に表示される走行状態解析画面は、下図に示すとおりであり、以下にその見方と解析内容を説明する。

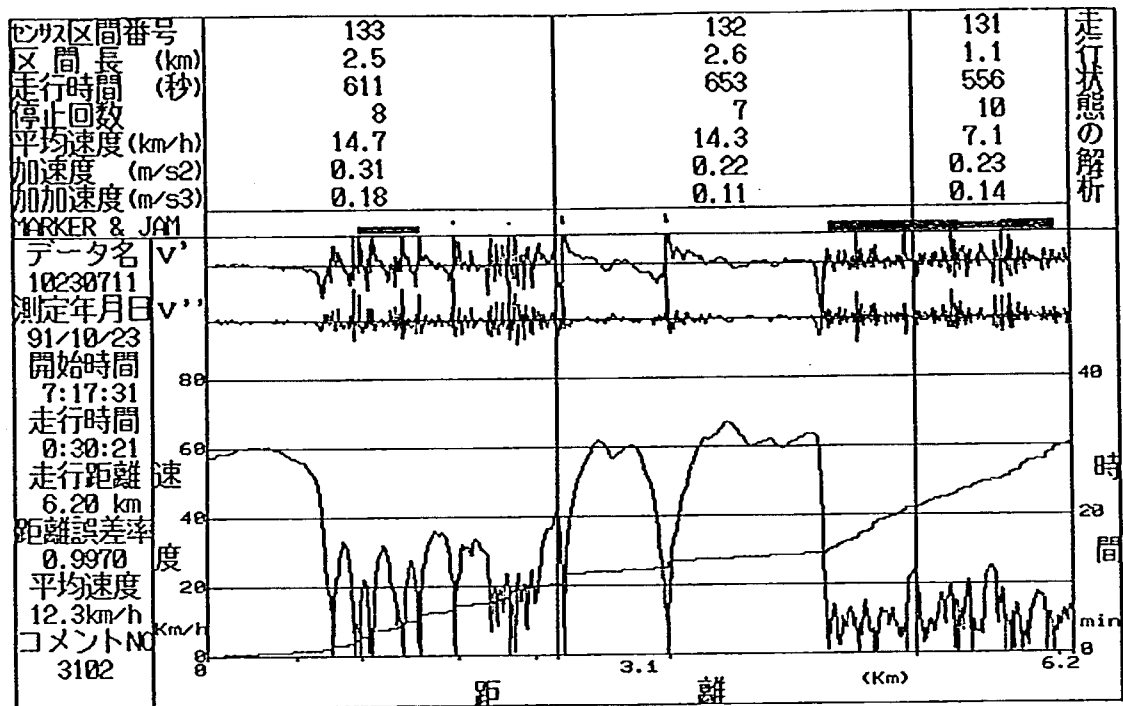


図3-21 走行状態図の図示例

(1) センサ区間番号

路線データから読み込まれたセンサ区間番号を示す。

(2) 区間長

路線データから読み込まれたセンサ区間の延長を示す。

(3) 走行時間

センサ区間毎の走行時間を秒単位で示す。

(4) 停止回数

走行速度をもとにして、停止回数を算定したもので、以下に述べる基準で判断をしている。

- ① 通常走行から4 km/h以下に速度がなった場合停止したと判断する。
- ② 停止状態から速度が上昇し、10 km/h以上になった時、停止状態が終了したと判断する。
- ③ 4 km/h以下に速度が低下し、10 km/h以上に速度が回復するまでの間は、1回の停止とする。

(5) 平均速度

センサス区間毎の平均速度（旅行速度）を示す。

(6) 加速度、加加速度

センサス区間毎の加速度及び加加速度の絶対値の平均値を示している。

(7) マーカ情報（MARKER）

測定時にマーカを入力した場合に、マーカ種別に従って色別に表示される。

交通事故地点	-----	白
道路工事地点	-----	黄色
幅員狭少、その他	-----	紫

(8) 渋滞情報（JAM）

平均速度及び瞬間最高速度をもとに、以下の条件によって渋滞区間を設定し、示したものである。なお、渋滞表示で太線は渋滞の激しいもの、細線は激しくないものを意味している。

〔渋滞判定の方法〕

- ① 走行状態での渋滞判定は、まず平均走行速度が20 km/h以下でかつ最高速度が35 km/h（住復2車線道路では30 km/h）以下の区間を渋滞区間の候補とする。
- ② 停止状態で、直前の区間が渋滞区間の候補となっている場合、当該停止区間も渋滞区間の候補とする。
- ③ 渋滞・非渋滞区間の各候補区間の区間長が300 m以下のものは、前後区間の状態と等しいとして補正を行い、渋滞区間を確定する。
- ④ 渋滞区間の候補の内、平均走行速度が15 km/h以下でかつ最高速度が20 km/h以下のものについては、渋滞の激しい区間とする。

(9) 走行状況図

時間-距離図、速度-距離図、加速度-距離図、加加速度-距離図を示す。

横軸に起点からの距離（km）が表示され、横軸下側に地点データ位地を点で示す。

左側縦軸に速度、右側縦軸に指定区間の始点からの積算走行時間（分）を示す。

中段には V' （加速度）及び V'' （加加速度）の変化を示す。加速度及び加加速度は、画面の1ドット当たりそれぞれ0.1 m/s²、0.1 m/s³である。なお、加速度、加加速度の基線間は、32ドットとなっている。

(10) データ名

旅行速度測定装置により自動的に作成される調査データファイル名である。

ファイル名の表すものについては、付録を参考にされたい。

(11) 測定月日、開始時間、走行時間

本調査1回の走行の測定月日と調査の開始時間（測定開始時間）、測定開始から測定終了までの走行時間を示す。

(12) 走行距離

路線データで作成された調査区間の全路線延長を示す。路線データが入力されない場合には、旅行速度測定装置の調査データからの実測距離を示す。

(13) 距離誤差率

実測距離と路線データの距離との比率を示したものであり、誤差が大きい場合はデータ類をチェックすることを勧める。

距離誤差率 = (路線データの延長 - 実測距離) / 路線データの延長

(14) 平均速度

調査区間全線の平均速度（旅行速度）を示す。

(15) コメントNO

旅行速度測定装置の操作により、入力されたコメントNoを示す。

(16) COPY(Y/N)

本表示画面の印刷に対しての問いである。Y（印刷実行）、N（印刷しない）の入力により次の操作へうつる。なおこの部分は画面に表示されるが印刷されない。

3. 3 波滯情報の集計

波滯情報の集計は、走行状態解析プログラムにより作成されたセンサス区間毎の集計データを読み込み集計するものである。

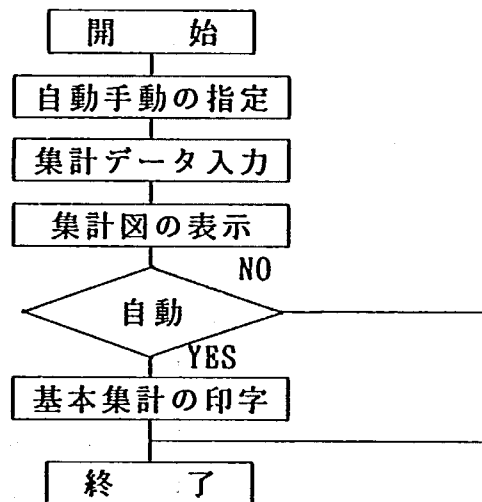


図 3 - 2 2 波滯情報の集計手順

波滯情報の集計では、集計用フロッピーに納められている全ての集計データ（拡張子に「.SUM」とついているもの）を読み込み、指定したセンサス区間毎に、上り下り別に旅行速度や波滯状況を集計するものである。

まず、集計を自動で行うか、手動で行うかを選択する。一般には、「自動」を選択する。手動にした場合は、集計結果の図を「RETURN」キーを押す毎に表示して見ることができるが、画面のハードコピーや、集計表の印字は行われない。そのため、手動集計は、簡単な集計チェックに利用できる。

つづいて、画面の指示に従って、路線番号及びセンサス区間番号を入力すると集計作業が開始され、自動を選択した場合には、結果の印字が自動的に行われる。

次に本集計プログラムにより集計され印刷される様式について示す。

1) 旅行速度日変動

調査日別の旅行速度のレンジ（最高、最低）及び平均値を上下別に示したものである。

2) 渋滞長日変動

調査日別の渋滞長のレンジ（最大、最小）及び平均値を上下別に示したものである。なお、平均値は、渋滞の発生した時間帯のみの平均を示している。

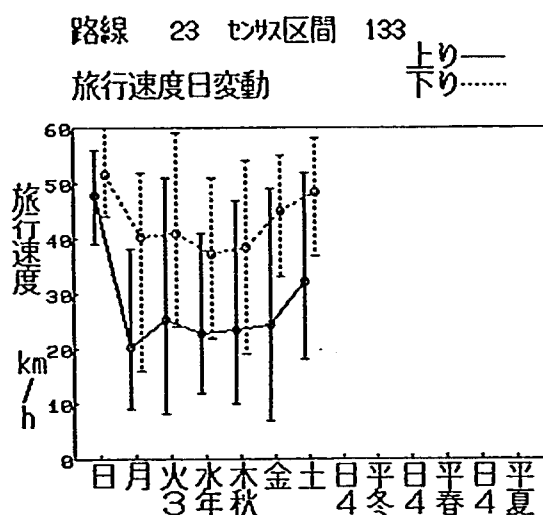


図 3-23 旅行速度日変動図

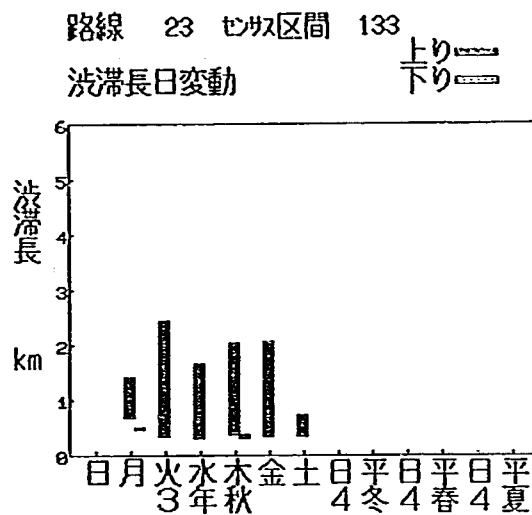


図 3-24 渋滞長日変動図

3) 旅行速度累加曲線図

指定したセンサス区間の全調査データから、旅行速度別の累加率を示したものである。

4) Q-V図

指定したセンサス区間の全調査データから、Q（交通量）とV（旅行速度）の関係をプロットしたものである。なお、Q（交通量）は、1車線当りの全車時間交通量を用いている。

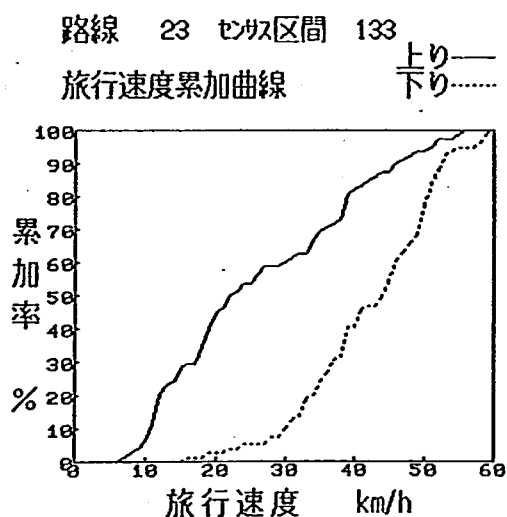


図 3-25 旅行速度累加曲線図

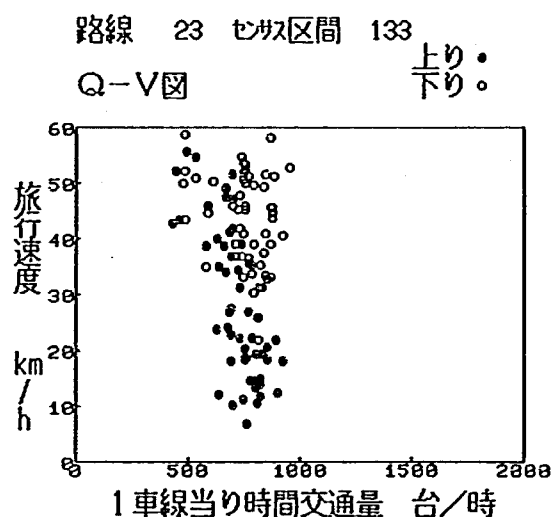


図 3-26 Q-V図

5) 旅行速度時間変動

1日の時間帯別旅行速度の変化を上下別に示したものである。1時間帯に複数のデータがある場合は、その平均値を示している。また、データのある時間帯には○印を付けてあり、データのない時間帯は、折れ線を飛ばして結んでいる。

6) 渋滞長時間変動

1日の時間帯別総渋滞長の変化を上下別に棒グラフで示したものである。

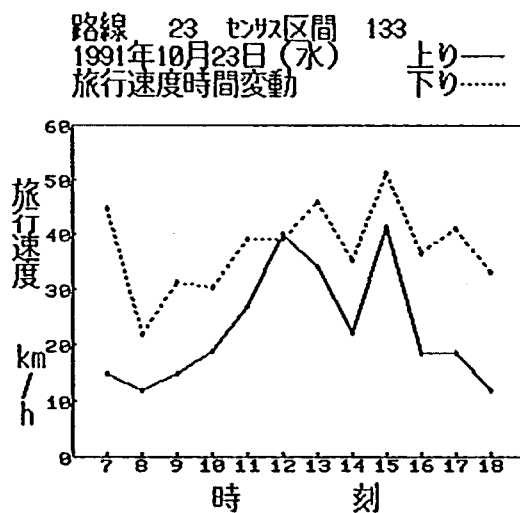


図3-27 旅行速度時間変動図

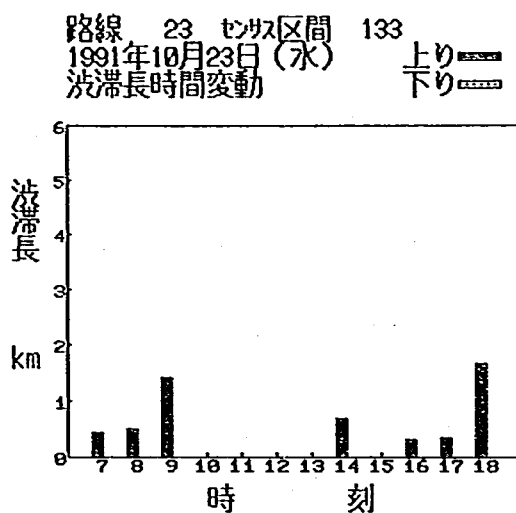


図3-28 渋滞長時間変動図

7) 渋滞状況図

1日の渋滞発生状況を示したものである。中央線に起点から1km毎のマークが入っているので渋滞の発生位置を把握できる。なお、表示する渋滞は、1時間帯に発生した渋滞長の長いものから3個までである。

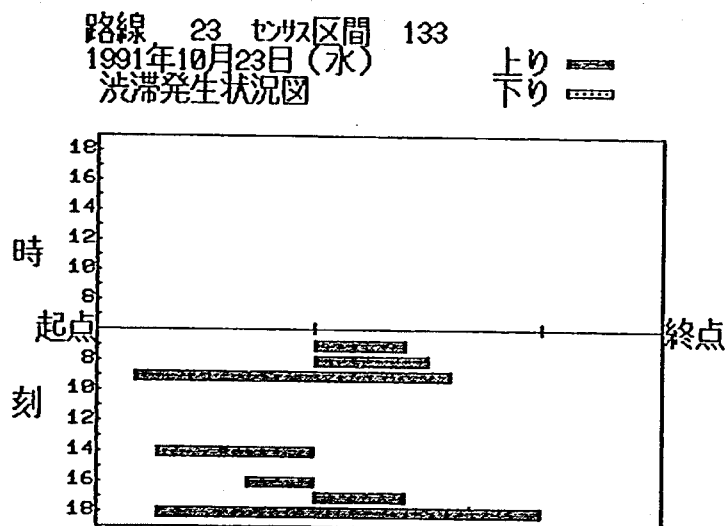


図3-29 渋滞状況図

8) 基本集計

指定したセンサス区間の走行状況を上下別・調査日時順に一覧表にしたものである。日平均値は、調査日別・上下別の平均値である。ただし、渋滞関連指標については、渋滞の発生したものの平均であり、1日の平均ではないので注意する。

表3-1 基本集計表出力結果

地域コード 23111 調査単位区間番号 13300 路線番号 23 道路種別 3 区間長 2500 ■ 車線数 3

調査方向	調査日	天候 視界 路面状況	通過時刻		交通量 (台/時)		放 行 時 間	停 止 回 数	放 行 速 度	平均 加 速 度	平均 加 速 度	停 止 時 間	総渋滞長 (10m)		渋滞区間 通過時間 (秒)		放 行 速 度 区 間 内	最大 渋滞長		
			起 点	終 点	全車	大型							ラン ク 1	ラン ク 2	ラン ク 1	ラン ク 2		0.1 km/h	100 m	100 m
年月日時			時分秒	時分秒	全車	大型	時分秒		0.1 km/h	0.01 m/s2	0.01 m/s3	秒	ラン ク 1	ラン ク 2	ラン ク 1	ラン ク 2	0.1 km/h	100 m	100 m	
1	911023	7 1 1 1	71731	72741	2385	0	01011	8 147	31	18	227	43	0	204	0	76	11	4		
2	911023	7 1 1 1	71143	715 3	2615	0	0 321	0 448	24	13	0	0	0	0	0	0	0	0		
1	911023	8 1 1 1	82440	83725	2456	0	01245	9 118	22	12	317	18	33	93	168	70	10	5		
2	911023	8 1 1 1	81017	817 9	2429	0	0 652	5 218	35	17	53	0	0	0	0	0	0	0		
1	911023	9 1 1 1	91639	92652	2310	0	01013	9 147	30	20	111	76	64	234	235	107	9	14		
2	911023	9 1 1 1	9 954	91443	2442	0	0 450	2 311	35	22	11	0	0	0	0	0	0	0		
1	911023	10 1 1 1	102039	102838	2248	0	0 759	5 188	27	19	133	0	0	0	0	0	0	0		
2	911023	10 1 1 1	10 859	101357	2370	0	0 458	3 302	40	25	31	0	0	0	0	0	0	0		
1	911023	11 1 1 1	111639	112214	2043	0	0 534	2 269	17	8	77	0	0	0	0	0	0	0		
2	911023	11 1 1 1	11 928	111318	2143	0	0 350	1 391	31	17	2	0	0	0	0	0	0	0		
1	911023	12 1 1 1	1224 6	122750	1873	0	0 345	1 401	23	14	17	0	0	0	0	0	0	0		
2	911023	12 1 1 1	121719	122110	2121	0	0 350	1 391	28	17	3	0	0	0	0	0	0	0		
1	911023	13 1 1 1	131142	1316 5	2003	0	0 424	2 341	27	13	54	0	0	0	0	0	0	0		
2	911023	13 1 1 1	13 655	131011	2242	0	0 316	0 459	25	19	0	0	0	0	0	0	0	0		
1	911023	14 1 1 1	143430	144112	2175	0	0 643	4 223	29	16	94	69	0	239	0	104	15	7		
2	911023	14 1 1 1	142356	142811	2463	0	0 415	1 353	23	15	2	0	0	0	0	0	0	0		
1	911023	15 1 1 1	152231	1526 9	2050	0	0 338	1 413	23	12	2	0	0	0	0	0	0	0		
2	911023	15 1 1 1	151735	152031	2219	0	0 256	0 512	21	10	0	0	0	0	0	0	0	0		
1	911023	16 1 1 1	161649	162458	2260	0	0 810	6 184	25	15	50	0	32	0	150	77	15	3		
2	911023	16 1 1 1	161122	161529	2298	0	0 4 6	1 365	24	14	36	0	0	0	0	0	0	0		
1	911023	17 1 2 1	171729	172537	2556	0	0 8 7	5 185	27	17	160	31	4	125	31	82	11	4		
2	911023	17 1 2 1	171028	1714 8	2534	0	0 340	0 410	19	18	0	0	0	0	0	0	0	0		
1	911023	18 1 3 1	1817 0	182931	2464	0	01230	8 120	23	15	267	163	3	534	20	108	5	17		
2	911023	18 1 3 1	1810 8	181440	2596	0	0 432	1 330	38	28	2	0	0	0	0	0	0	0		
小計(日平均値)			上り		2235	0	0 850	5 228	25	15	126	57	19	204	86	89				
			下り		2373	0	0 412	1 374	29	18	12	0	0	0	0	0				

付 録

1. 道路データ、路線データの記録様式
2. 調査データの記録様式
3. 漢字入力の方法
4. スクリーンエディタの使い方
5. プログラムリスト
6. 旅行速度常時観測調査 実施要領
7. 旅行速度測定装置の概要

1. 道路データ、路線データの記録様式

路線データの作成で述べた各データは、以下に示す拡張子で区別されている。

- ・道路データ : 「XXXXXXX.ORG」
- ・路線データ : 「XXXXXXX.ROD」

各データの記録様式及び内容については、以下のとおりであり、修正、データチェックの参考にされたい。

(1) 道路データ

① 路線番号

路線毎に定めた番号を示す。(最大4桁の英数字)

② センサス区間数

当該路線に存在するセンサス区間の数を示す。センサス区間をさらに分割している場合は、その分割数も考慮した総合計となる。なお、この区間数と等しい数のセンサス区間データが必要となる。1路線のセンサス区間数は、最大50区間である。

センサス区間	0010	0011	0022		
分割区間			01	02	03

上記例では、センサス区間数は、「5」となる。

③ 路線名称

漢字で20桁(半角文字で40桁)以内の正式な路線名称である。

④ センサス調査区間番号

センサス調査区間番号の他に、都道府県コード、市町村コード及びセンサス区間を分割している場合の分割番号である。なお、センサス区間を分割していない場合は、「00」となる。

⑤ 区間延長

当該センサス区間（分割の場合は、分割区間）の延長を10m単位で示す。すなわち、区間長が1.5kmの場合は、「150」となる。

⑥ 道路種別

以下の道路種別コードを示す。

- 1：高速自動車国道
- 2：都市高速道路
- 3：一般国道
- 4：主要地方道（都道府県道）
- 5：主要地方道（指定市市道）
- 6：一般都道府県道

⑦ 車線数

当該センサス区間の平均車線数（片側）を示す。

⑧ 地点数

当該センサス区間に含まれる主要地点あるいは信号交差点の数。ここで表示した地点数分の地点データが必要となる。なお、信号交差点については、押しボタン式を除くすべてを取り上げることが解析する上で望ましい。1センサス区間に含まれる地点数は、最大50地点とする。

⑨ キロポスト

対象地点のキロポストを10m単位で示す。

⑩ 地点番号

センサス区間において上りから順に付けられる地点番号である。なお、道路データ作成プログラムを用いた場合、地点番号は、自動的につけられる。

⑪ 地点名称・交差点名称

地点名称または、交差点名称を示す。なお、複数の路線が同一交差点を通過している場合、名称は統一する。

〔道路データファイル〕

① 路線名データ

路線番号	センサス区間数	路線名称
4	2	20

② センサス区間データ

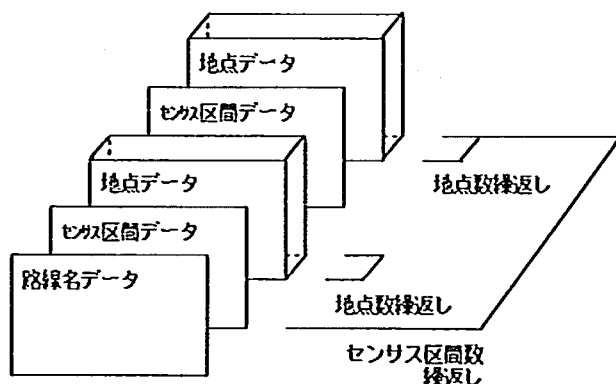
センサス調査区間						区間延長 (10m)	道路種別	車線数	地点数
都道府県	市区町村	センサス区間番号	分割番号						
2	3	4	2			4	1	1	2

③ 地点データ

キロポスト (10m)	地点番号	地点名称・交差点名称
5	2	22

図－1 道路データの記録様式

データの並び



002303国道23号
 2311101310001103303
 3599001築地インター東
 3600002築地インター西
 3610003寛政インター
 2311101320002603303
 3610001寛政インター
 3628002十一屋交差点
 3636003宝神交差点
 2311101330002503306
 3636001宝神交差点
 3637502庄内新川橋東交差点
 3643003庄内新川橋西交差点
 3650004藤前西交差点（信号無）
 3654505流通団地交差点（インタ）
 3661006名古屋市・飛島村境

(2) 路線データ

① センサス区間数

走行調査を行うルートに含まれるセンサス区間（分割の有無には関係ない）の総数であり、この数だけセンサスデータが必要となる。センサス区間数の最大は、50区間とする。

② 調査方向

調査方向については、上りを「1」、下りを「2」とする。ここで、上りはキロポストが減少する方向とする。

③ 累積距離

走行調査を行うルートの起点からの累積距離で10m単位で示す。

その他の項目については、道路データと同様の内容である。

【路線データファイル】

① 区間数データ

セ ン サ ス 区 間 数				
1	2	3	4	5
		5		

② センサスデータ

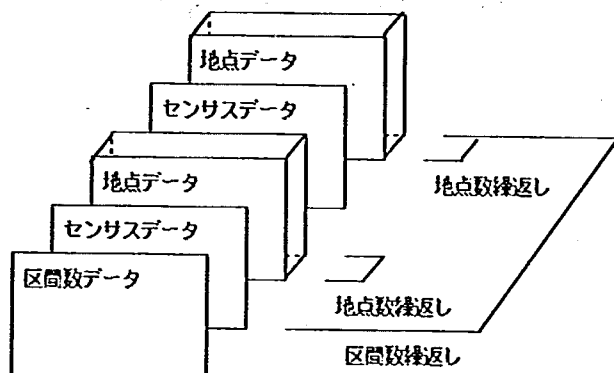
センサ区間番号											路線番号				調査方向	車線数	地点数	道路種別	
都府	道県	市町	区	村	センサ区	センサ区	番号	番号	分割	割	号								
2		3					4		2			4				1	1	2	1
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

③ 地点データ

キ ロ ボ ス ト (10m)	累 加 距 離 (10m)	交 差 点 ・ 名 称
5	5	22
1 2 3 4 5	6 7 8 9 10	11 12 13 14 15 16 17 18 19 20 21 22 23 30 31 32

図-2 路線データの記録様式

データの並び



3
23111013300002313 63
36610 0名古屋市・飛島村境
36545 65流通団地交差点(インタ)
36500 110藤前西交差点(信号無)
36430 180庄内新川橋西交差点
36375 235庄内新川橋東交差点
36360 250宝神交差点
23111013200002313 33
36360 250宝神交差点
36280 330十一屋交差点
36100 510寛政インター
23111013100002313 33
36100 510寛政インター
36000 610築地インター西
35990 620築地インター東

(下り方向の路線データ)

2. 調査データの記録様式

MS-DOSでフォーマットされ、以下に示す拡張子で区別されている。

- ・ヘッダーファイル：「*****.HED」
- ・ステップデータ：「*****.STP」

(1) ファイル名について

積算スタートを開始した「月、日、時、分」（7桁、分については十分単位であることに注意）と「積算スタート回数-1」（1桁、最大値9）の8桁がファイル名となる。

積算スタート回数は、旅行速度測定装置から、フロッピーにデータをセーブ、又はメモリークリアを押してから測定回数を表す。

例)

1990年10月21日 8時3×分に2回目に計測したデータが下記のように設定されファイル名となる。（×についてはファイル名にならない）

1021083 HED
1021083 STP

(2) ヘッダーファイル(*****.HED)

ヘッダーファイルは、一回の積算スタート毎に作成される。1byte目より次の様な内容で書き込まれる。

ファイルNo. 設定	:	1 byte (BCD表現)
コメントNo. 設定	:	2 byte (BCD表現)
日付データ	:	3 byte (各2桁、BCD表現)
開始時刻	:	3 byte (各2桁、BCD表現)
終了時刻	:	3 byte (各2桁、BCD表現)
サンプル間隔設定値	:	1 byte (BIN表現)
基準タイヤ設定値	:	2 byte (BIN表現)
補正タイヤ設定値	:	2 byte (BIN表現)
積算走行距離データ	:	4 byte (BIN表現)
積算走行時間データ	:	2 byte (BIN表現)
ステップ数	:	2 byte (BIN表現)

以上、全25byteより構成されている。

旅行速度測定装置内のメモリーには最大全10回計測(250byte)まで記憶可能である。

(3) ステップデータ (*****.STP)

サンプル間隔設定値により、一定距離毎に次の様なデータが書き込まれる。

積算走行距離	:	2 b y t e (	BIN表現	)
ステップ所要時間	:	2 b y t e (	BIN表現	)
ステップ終了時刻	:	3 b y t e (	各 2 桁、BCD表現	)
マーカ信号	:	1 b y t e (	BIT対応	)

以上 8 b y t e / ステップで構成されている。

旅行速度測定装置のメモリーには最大 5 0 0 0 0 ステップ (4 0 0 k b y t e)
まで記憶可能である。

ステップデータは各積算スタートファイル毎に、可変長である。

今回 1 ステップを 1 m としているので、50,000m までフロッピーにセーブせず測定可能である。

3. 漢字入力の方法

漢字の入力は、種々日本語入力フロントプロセッサがあるが、下記には、銆ジヤストシステムの「A T O K 6」日本語ワードプロセッサ「一太郎」で使われているものを利用する場合について示す。

- ① 漢字入力を行う必要がある場合、[CTRL] + [XFER] を押す。これにより最下段右端に「連ローマ字漢字」の表示が現れ、ローマ字漢字変換が可能となる。
- ② 日本語の変換は、ローマ字にて日本語を入力すると最下段左端にひらがなが表示される。単語の句切りで[スペース]キーを押すと漢字に変換される。この漢字が正しい場合は、[リターン]キーによりデータの入力欄に漢字が移動する。正しくない場合は、更に[スペース]キーを押すことによって候補の漢字が表示されるので、番号により選択する。
- ③ 特殊な変換キーとして以下のものがあるので、必要に応じて利用する。
 - f 6 : ひらがな変換
 - f 7 : カタカナ変換
 - f 8 : 半角変換
 - f 9 : 無変換（入力したまま）
 - f 10 : 入力モードの変換
- ④ 漢字入力が終了した場合、[CTRL] + [XFER] を押すと元の状態に戻り、半角文字による数値、アルファベットの入力状態となる。

4. スクリーンエディタの使い方

道路データ、路線データ、集計データをプログラムを利用せずに直接作成、修正したい場合には、市販のエディタやワープロソフトが利用できる。以下にスクリーンエディタ（MIFES）を使用した場合について示す。

① ファイル名の指定

エディタを起動すると、図-3に示すようなファイル名を指定するメニューが表示される。

窓の中に、データディスクのある装置番号を指定する。例えば、装置Bにデータファイルがある場合、[B*,*]と入力する。これによって、「カレントディレクトリー」が「B:」と代わり、下の窓に装置Bに入っているファイルの一覧表が表示される。

ファイルの指定は、矢印キーを用いて修正したいファイルを選択することで行える。

編集したいファイル名を入力して下さい [テキスト番号 : 0]			
B:*,*			
カレントディレクトリー = A : ¥ PATH = B:*,*			
23. ROD	111. ORG	222. ORG	CHUBU. TXT

図-3 ファイル名の指定例

② エディターによる入力

指定されたファイルを読み込み、図-4に示す画面となる。データの入力は、矢印キーによりカーソルを移動して、必要なデータをタイプすることで行う。

この際、よく利用するキーは、以下のとおりである。

[INS] : このキーを押すことにより挿入と上書きを切り替えることができる。

[DEL] : カーソル位置の文字を消去する。

B:¥23.ROD	[1: 1]	<挿入>	CTRL+Jでヘルプ
-----------	---------	------	------------

```

002303国道23号
2311101310001103303
3599001築地インター東
3600002築地インター西
3610003寛政インター
2311101320002603303
3610001寛政インター
3628002十一屋交差点
3636003宝神交差点
2311101330002503306
3636001宝神交差点
3637502庄内新川橋東交差点
3643003庄内新川橋西交差点
3650004藤前西交差点(信号無)
3654505流通団地交差点(インタ)
3661006名古屋市・飛島村境
[EOF]
-
-
-
-
MENU1 MENU2 MENU3 SPLIT FIND SEL CUT COPY PASTE TAGJR

```

図-4 エディター画面例

なお、漢字の入力については、前述のとおりである。

③ データの保存

データの修正が終了し、ファイルに保存する場合は、[f 1]キーを押してメニューを出し、データを保存する。メニューは、図-5に示すとおりであるが、一般に利用される項目は、

E: データを保存してエディタを終了する

Q: データを保存せずにエディタを終了する。

の2種類である。データの修正を中断して(修正を行わない)終了する場合のみ「Q」を選択し、その他の場合は「E」を選択するとよい。選択は、矢印キーで行える。

D:¥23.ROD		[1: 1]	<挿入>
FILE/QUIT		:ファイルの関係と操作	
E	Save & Quit	:現テキストのセーブと編集終了	
S	Save	:現テキストのセーブ 編集は継続	
Q	Quit	:現テキストの強制的な編集終了	
O	Original file	:現テキストの編集を最初からやり直す	
D	Duplioato	:現テキストの二重化	
R	Rename	:現テキストのパス名の変更	
A	All Save & End	:現テキストのセーブとエディタ終了	
O		:強制的なエディタの終了	
I	Insert file	:カーソル行の前にファイルを読む込む	
N	New file	:新しいファイルの編集	
		:	

図-5 データ保存のメニュー

5. プログラムリスト

旅行速度常時観測調査集計プログラムのフロッピーには以下プログラムが入っている。これらは、実行可能ファイルの形で入っているためプログラムの中身を見たり、修正することは出来ない。

RODF. EXE	路線データの作成
PRRD. EXE	路線データの印字
TRA2. EXE	走行状態の解析
JAMT. EXE	渋滞情報の集計

本プログラムリストは解析上の主要部分である以下2つのソースリストを掲載している。

- ・ 走行状態の解析プログラム

TRA2. BAS ----- TRA2. EXEのソースプログラム

- ・ 渋滞情報の集計プログラム

JAMT. BAS ----- JAMT. EXEのソースプログラム

なお本プログラムはMicrosoft QuickBASICにより作成したものである。

TRA2. BAS

```

1: '*****
2: '*   旅行速度測定装置による走行解析   *
3: '*   渋滞判断、加速度付き             *
4: '*   常時観測集計用                   *
5: '*   BY Y.YOSHIDA (YEC)   Sep., 1991   *
6: '*****
7: '----- サブルーチン定義 -----
8: DECLARE SUB MK.FIL (NPSC&)
9: DECLARE SUB RE.CHK (LDIS&, NPSC&)
10: DECLARE SUB MOV.DAT (II&, NPSC&)
11: DECLARE SUB RE.CHK2 (I&, LDIS&, NPSC&)
12: DECLARE SUB DISP.RES ()
13: DECLARE SUB SELE.FIL (a$, U$, f$)
14: DECLARE SUB PRN.FIL (N&, ATT%, FLN$)
15: DECLARE SUB SEL.FIL (PATH$, NF&, FLN$(), ATT%())
16: DECLARE SUB CUR (NF&, FLN$(), ATT%(), NN&)
17: DECLARE SUB DSP.STAT (KDS&, KDE&)
18: DECLARE SUB SET.TIME ()
19: DECLARE SUB SET.W ()
20: DECLARE FUNCTION ADJ.ROAD! (COEF)
21: DECLARE SUB STEP.OPEN ()
22: DECLARE SUB DISP.HEAD ()
23: DECLARE SUB INIT.CON ()
24: DECLARE SUB ROAD.READ ()
25: DECLARE SUB STEP.READ (II&)
26: DECLARE SUB HEAD.READ ()
27: DECLARE SUB DISP.AXIS ()
28: DECLARE FUNCTION CHK.FILE% (PATH$)
29: '$INCLUDE: 'YOSHIDA.BI'
30: '----- グローバル変数定義 -----
31: DEFDBL T: DEFINT I-N: DEFINT X-Y
32: DIM SHARED NA.ST$(100), DIS.ST(100), PS(100, 2), PC(100)
33: DIM SHARED ROAD.NO(30), HOUKOU(30), WW(30), NO.st(30, 2), CLASS$(30)
34: DIM SHARED CITY.NO(30), SENS.NO(30), FHOR(30, 2), FMIN(30, 2), FSEC(30, 2)
35: DIM SHARED P.TIM(100, 2), P.MAX(100), P.MRK(100, 8), P.STP(100)
36: DIM SHARED LIMIT(4, 2), TIT$(8), MK(8), TRAFFIC(2), WHETHER(3)
37: DIM SHARED KBOD(30) AS LONG, T.CON(30), N.CON(30), S.CON(30), F.CON(30)
38: DIM SHARED SET.TIM(30, 2), KBPOS(30), TOTA(30), TOTA2(30), NTOT(30)
39: COMMON SHARED /CCC1/ NA.file$, COMENT, Year, Month, Day, Shor, Smin, Ssec
40: COMMON SHARED /CCC2/ Ehor, Emin, Esec, Dstep, Stime, Nstep, COEF
41: COMMON SHARED /CCC3/ t1, t2, T3, TD, TIM, DIST, SH, SM, SS, AA$, DEF.DIS
42: COMMON SHARED /CCC4/ ROAD.SW, LINE.SW, CONG.SW, NO.file, NO.sen, NO.stt
43: COMMON SHARED /CCC5/ HEAD.fil$, STEP.fil$, CONG.fil$, ROAD.fil$, TEMP.fil$, IN
    EN%
44: COMMON SHARED /CCC6/ L.V1, L.V2, L.VM1, L.VM2, W
45: COMMON SHARED /CCC7/ Mstart, Mend, Mstart0, Mend0, DMAX, TBASE
46: COMMON SHARED /CCC8/ DSP, OFFSET, ALDST, Unit$, Unitr$, JAM.SW
47: '----- 初期設定 -----

```

```

48: Unit$ = "A:": Unitr$ = "B:"
49: START:
50: FOR I = 1 TO 30
51:     KBOD(I) = 99999!: T.CON(I) = 0: N.CON(I) = 0
52:     S.CON(I) = 0: F.CON(I) = 0
53:     SET.TIM(I, 1) = 0: SET.TIM(I, 2) = 0: KBPOS(30) = 0
54:     TOTA(I) = 0: TOTA2(I) = 0: NTOT(I) = 0
55: NEXT
56: '----- 渋滞判定基準 -----
57: RESTORE JAMDATA
58: FOR J = 1 TO 2: FOR I = 1 TO 4: READ LIMIT(I, J): NEXT I, J
59: JAMDATA: DATA 20, 30, 15, 20, 20, 35, 15, 20
60: L.D = 300
61: L.V1 = LIMIT(1, 2): L.VM1 = LIMIT(2, 2)
62: L.V2 = LIMIT(3, 2): L.VM2 = LIMIT(4, 2)
63:
64: '----- メインルーチン開始 -----
65:     CALL INIT.CON
66:     SCREEN 88: CLS : CALL DISP.AXIS: CALL DISP.HEAD
67:     Mstart = 0: Mend = INT(DMAX)
68:     DIST = Mend - Mstart: DP = 500 / DIST
69:     Mstart0 = Mstart: Mend0 = Mend
70:     CALL SET.TIME
71:     CALL STEP.OPEN: TH = Shor: TM = Smin: TS = Ssec
72:     GOSUB DISP.VD
73:     LOCATE 25, 1: PRINT "COPY (Y/N)";
74:     YN$ = ""
75:     WHILE YN$ = ""
76:         YN$ = INKEY$
77:     WEND
78:     LOCATE 25, 1: PRINT SPACES$(10);
79:     IF YN$ = "y" OR YN$ = "Y" THEN
80:         IF ROAD.SW = 1 AND JAM.SW = 1 THEN CALL MK.FIL(NPSC)
81:         CALL Hcopy(7, 0)
82:     END IF
83:     CLS : COLOR 6
84:     INPUT "次の調査データを解析しますか (y/n) "; YN$
85:     IF YN$ = "y" OR YN$ = "Y" THEN
86:         GOTO START
87:     ELSE
88:         DO WHILE (1)
89:             PRINT : PRINT "プログラムディスクを装置Aに挿入して下さい"
90:             PRINT "    確認したら[RETURN]を押して下さい "
91:             WHILE INKEY$ = "": WEND
92:             a$ = "A:TRA2.EXE": ISW% = CHK.FILE$(a$)
93:             IF ISW% THEN EXIT DO
94:             BEEP
95:         LOOP
96:         CLS : END

```

```

97:         END IF
98:
99: '----- メインルーチン終了 -----
100:
101: DISP.VD: '-- 走行図作成サブルーチン --
102:
103: '----- MEMORY CLEAR ----
104: TOTA(1) = 0: TOTA2(1) = 0: NTOT(1) = 0
105: N.CON(1) = 0: T.CON(1) = 0
106: NSTOP = 0: ISWSTP = 0
107: C.C = 0: C.L = 0: C.T = 0: C.MAX = 0: C.ST = NST: NPSC = 1
108: SENS.N = 1
109: FOR I = 1 TO 100: P.MAX(I) = 0: P.STP(I) = 0
110:     FOR J = 1 TO 8: P.MRK(I, J) = 0
111: NEXT J, I
112:
113: NST = 1: NED = Nstep: YJAM = 121
114: DSP = DP * COEF: OFFSET = 111 - NST * DSP
115: DSP.STP = 1 / DSP: ISWMARK = 1
116:     FOR II = NST TO NED: CALL STEP.READ(II)
117:         IF II = NST THEN
118:             PS(NPSC, 1) = NST: P.TIM(NPSC, 1) = 0
119:             C.TIM = 0: SET.TIM(1, 1) = 0
120:             IF NST = 1 THEN PS(NPSC, 1) = 0
121:             FHOR(SENS.N, 1) = SH: FMIN(SENS.N, 1) = SM: FSEC(SENS.N, 1) =
SS
122:         END IF
123:         IF II = NED THEN
124:             PS(NPSC, 2) = NED: P.TIM(NPSC, 2) = TO + T
125:             SET.TIM(SENS.N, 2) = TO + T
126:             FHOR(SENS.N, 2) = SH: FMIN(SENS.N, 2) = SM: FSEC(SENS.N, 2) =
SS
127:         END IF
128:         IF NO.sen = 1 THEN GOTO 1680
129:         IF II * COEF > KBOD(SENS.N) THEN
130:             FHOR(SENS.N, 2) = TH: FMIN(SENS.N, 2) = TM: FSEC(SENS.N, 2) =
TS
131:             SENS.N = SENS.N + 1
132:             FHOR(SENS.N, 1) = SH: FMIN(SENS.N, 1) = SM: FSEC(SENS.N, 1) =
SS
133:             TOTA(SENS.N) = 0: TOTA2(SENS.N) = 0: NTOT(SENS.N) = 1
134:             SET.TIM(SENS.N - 1, 2) = TO + T
135:             SET.TIM(SENS.N, 1) = TO + T
136:             IF ROAD.SW = 1 THEN W = WW(SENS.N): CALL SET.W
137:         END IF
138: 1680 '-----速度計算-----
139:         TLIMIT = 1!
140:         TX = (SH - TH) * 3600 + (SM - TM) * 60 + (SS - TS)
141:         IF II = NST THEN
142:             V = 3.6 * COEF / TIM * 10000: Y0 = 370 - V * 2

```

```

143:                                V0 = V: V1 = V: V2 = V: T0 = 0: X0 = 111: x1 = 111:
    y1 = 370
144:                                TX = 0: TSUM = 0: NSUM = 0: T.ALL = 0
145:                                ACC = 0: ACV = V: x2 = 111: y2 = 144
146:                                TOTAC = 0: TOTAC2 = 0: NTOTA = 0
147:                                ACV1 = 0: X3 = 111: Y3 = 176
148:                                END IF
149:                                T = TX: IF TX < 5 THEN T = TIM / 10000
150:                                IF T = 0 THEN T = TX
151:                                T.CON(SENS.N) = T.CON(SENS.N) + T * 100: T.ALL = T.ALL + T
152:                                IF ISWSTP = 1 THEN S.CON(SENS.N) = S.CON(SENS.N) + T * 100
153:                                TSUM = TSUM + T: NSUM = NSUM + 1
154:                                IF TSUM > TLIMIT THEN
155:                                    IF TSUM = 0 THEN V = V2 ELSE V = 3.6 * COEF * NSUM / TSUM
156:                                    VV = V
157:                                    T0 = T0 + TSUM: C.T = C.T + TSUM: C.L = C.L + NSUM: CTT = C.T
    - TSUM
158:                                    IF C.MAX < VV THEN C.MAX = VV
159:                                    X = 11 * DSP + OFFSET: Y = 370 - VV * 2: IF 11 = NST THEN YO =
    Y
160:                                    IF LINE.SW <> 1 THEN LINE (X0, Y0)-(X, Y), 7
161:                                    X0 = X: Y0 = Y
162: '----- 加速度を一定時間集計して算定 -----
163: '----- 速度も一定時間集計-----
164:                                    NTOT(SENS.N) = NTOT(SENS.N) + NSUM: NTOTA = NTOTA + NSUM
165:                                    ACC = (VV - ACV) / TSUM / 3.6
166:                                    TOTAC = TOTAC + ABS(ACC) * TSUM: TOTA(SENS.N) = TOTA(SENS.N) +
    ABS(ACC) * TSUM
167:                                    Y = 144 - ACC * 10: LINE (x2, y2)-(X, Y), 5
168:                                    x2 = X: y2 = Y: ACV = VV
169:                                    ACC1 = (ACC - ACV1) / TSUM
170:                                    TOTAC2 = TOTAC2 + ABS(ACC1) * TSUM: TOTA2(SENS.N) = TOTA2(SENS
    .N) + ABS(ACC1) * TSUM
171:                                    Y = 176 - ACC1 * 10: LINE (X3, Y3)-(X, Y), 5
172:                                    X3 = X: Y3 = Y: ACV1 = ACC
173:                                    TSUM = 0: NSUM = 0
174:                                END IF
175: 1800 '-----積算時間表示-----
176:                                IF LINE.SW = 2 THEN 1840
177:                                Y = 370 - T0 / TBASE * 160
178:                                LINE (x1, y1)-(X0, Y), 7: x1 = X0: y1 = Y
179: '-----マ-カ表示-----
180: 1840 FOR KK = 1 TO 8
181:                                IF MK(KK) <> 0 THEN
182:                                    Y = 113: LINE (X, Y)-(X, Y + 4), 8 - KK
183:                                    P.MRK(NPSC, KK) = P.MRK(NPSC, KK) + 1
184:                                    F.CON(SENS.N) = 1
185:                                END IF
186:                                NEXT KK
187: '-----停止回数カウント及び渋滞表示-----

```

```

188:      IF ISWSTP = 0 AND V > 4 THEN 2210
189:      IF ISWSTP = 1 THEN 2100
190:      '---- [RUN] -> [STOP] ----
191:      ISWSTP = 1: NSTOP = NSTOP + 1: P.STP(NPSC) = P.STP(NPSC) + 1
192:      N.CON(SENS.N) = N.CON(SENS.N) + 1
193:      IF C.T = 0 THEN 2070
194: 1930  LL = (C.L - 1) * COEF: VEL = 3.6 * LL / CTT: PC(NPSC) = 0: PS(NPSC, 1)
      = C.ST: PS(NPSC, 2) = II - 1
195: 1940  P.TIM(NPSC, 1) = C.TIM: P.TIM(NPSC, 2) = TO - T: P.MAX(NPSC) = C.MAX
196:      IF VEL > L.V1 OR C.MAX > L.VM1 THEN 2010
197:      K.JUM = 1: IF VEL <= L.V2 AND C.MAX <= L.VM2 THEN K.JUM = 2
198: 1970  PC(NPSC) = K.JUM
199: 1980  XA = C.ST * DSP + OFFSET: Y = YJAM
200:      XB = (II - 1) * DSP + OFFSET
201: 1990  IF K.JUM = 2 THEN LINE (XA, Y)-(XB, Y + 5), 2, BF
202:      IF K.JUM = 1 THEN LINE (XA, Y + 1)-(XB, Y + 4), 2, BF
203: 2010  IF NPSC = 1 THEN 2070
204:      IF PC(NPSC - 1) <> PC(NPSC) THEN 2070
205:      PS(NPSC - 1, 2) = II - 1: NPSC = NPSC - 1: P.TIM(NPSC, 2) = TO - T
206:      P.STP(NPSC) = P.STP(NPSC) + P.STP(NPSC + 1): P.STP(NPSC + 1) = 0
207: 2050  FOR KK = 1 TO 8
208:      P.MRK(NPSC, KK) = P.MRK(NPSC, KK) + P.MRK(NPSC + 1, KK)
209:      P.MRK(NPSC + 1, KK) = 0
210:      NEXT
211: 2060  IF P.MAX(NPSC) < C.MAX THEN P.MAX(NPSC) = C.MAX
212: 2070  C.ST = II: C.L = 1: C.T = 0: C.MAX = 0: NPSC = NPSC + 1: C.TIM = TO
213: 2080  GOTO 2210
214:      '---- [STOP] -> [RUN] ----
215: 2100  IF V <= 10 THEN 2210
216: 2110  ISWSTP = 0
217: 2120  PS(NPSC, 2) = II - 1: PS(NPSC + 1, 1) = II
218:      P.TIM(NPSC, 2) = TO - T: P.TIM(NPSC + 1, 1) = TO
219: 2130  P.MAX(NPSC) = C.MAX
220: 2140  IF NPSC = 1 THEN 2160
221: 2150  PS(NPSC - 1, 2) = II - 1: PS(NPSC, 1) = II
222:      P.TIM(NPSC - 1, 2) = TO - T: P.TIM(NPSC, 1) = TO
223: 2160  IF C.C = 0 THEN 2200
224: 2170  C.C = 0: XA = C.ST * DSP + OFFSET: Y = YJAM
225:      XB = (II - 1) * DSP + OFFSET
226: 2180  IF K.JUM = 2 THEN LINE (XA, Y)-(XB, Y + 5), 2, BF
227: 2190  IF K.JUM = 1 THEN LINE (XA, Y + 1)-(XB, Y + 4), 2, BF
228: 2200  C.ST = II: C.L = 1: C.T = 0: C.MAX = 0: C.TIM = TO
229: 2210  '----- END OF DISPLAY -----
230:      TH = SH: TM = SM: TS = SS: VO = V1: V1 = V2: V2 = V
231:      NEXT II: CLOSE #2: PS(NPSC, 2) = NED
232:      LOCATE 8, 71: PRINT USING "###.###": TOTAC2 / T.ALL;
233:      CALL DISP.RES
234:      PC(NPSC) = 0: IF PC(NPSC - 1) <> 0 THEN 2270
235:      PS(NPSC - 1, 2) = PS(NPSC, 2)
236:      P.TIM(NPSC - 1, 2) = P.TIM(NPSC, 2): P.MAX(NPSC) = C.MAX

```

```

237:         NPSC = NPSC - 1
238:
239: 2270 '--- RE-CHECK OF CONGESTION -----
240:         CALL RE.CHK(L.D, NPSC)
241:         Y = YJAM: LINE (112, Y)-(610, Y + 5), 0, BF
242:         FOR I = 1 TO NPSC: Y = YJAM: K.JUM = PC(I)
243:             IF K.JUM <> 0 THEN
244:                 XA = PS(I, 1) * DSP + OFSET: XB = PS(I, 2) * DSP + OFSE
T
245:                 IF K.JUM = 2 THEN LINE (XA, Y)-(XB, Y + 5), 2, BF
246:                 IF K.JUM = 1 THEN LINE (XA, Y + 1)-(XB, Y + 4), 2, B
F
247:             END IF
248:         NEXT I
249:         IF ROAD.SW <> 0 THEN
250:             DSTART = INT(Mstart / 100 + .5) / 10: DEND = INT(Mend / 100 + .
5) / 10
251:             FOR I = 1 TO NO.sen
252:                 SS = DIS.ST(NO.st(I, 1)): EE = DIS.ST(NO.st(I, 2))
253:                 IF SS <= DEND * 1000 AND EE >= DSTART * 1000 THEN
254:                     IF SS >= DSTART * 1000 THEN
255:                         XS = (SS - DSTART * 1000) * .5 / (DEND
- DSTART) + 111
256:                         LINE (XS, 370)-(XS, 0), 3
257:                     END IF
258:                     IF EE <= DEND * 1000 THEN
259:                         XE = (EE - DSTART * 1000) * .5 / (DEND
- DSTART) + 111
260:                         IF I <> NO.sen THEN LINE (XE, 370)-(XE,
0), 3
261:                     END IF
262:                 END IF
263:             NEXT
264:         END IF
265:
266: RETURN
267:
268: DEFSNG I-M, X-Y
269: FUNCTION ADJ.ROAD (COEF)
270: '--- 路線データによる距離補正 ---
271: IF ROAD.SW = 1 THEN
272:     T.DISTANCE = DIS.ST(NO.stt)
273:     DEF.DIS = TD / T.DISTANCE
274:     ADJ.ROAD = COEF * T.DISTANCE / TD: TD = T.DISTANCE
275: ELSE
276:     ADJ.ROAD = COEF
277:     DEF.DIS = 1
278: END IF
279: END FUNCTION
280:

```

```

281: DEFNG I-M
282: FUNCTION CHK.FILE% (PATH%)
283: '--- ファイル有無のチェック ---
284: DIM DTA AS DtaType
285: FUNC% = 0
286: f$ = GetFile$(PATH$, ATR%, DTA, FUNC%)
287: IF (f$ = "") THEN CHK.FILE% = 0 ELSE CHK.FILE% = -1
288: END FUNCTION
289:
290: SUB CUR (NF, FLN$( ), ATT%( ), NN)
291: '----- ファイル選択カーソル移動 -----
292:   NN = 1
293: JP.CUR1: CALL PRN.FIL(-NN, ATT%(NN), FLN$(NN))
294:           NO = NN: LXO = LX: LYO = LY
295:
296: DO: a$ = INKEY$: LOOP WHILE a$ = ""
297:   IF a$ = CHR$(27) THEN CLS : END
298:   IF a$ = CHR$(13) THEN GOTO JP.CUR2
299:   IF a$ = CHR$(0, 77) THEN NN = NN + 1: IF NN > NF THEN NN = 1
300:   IF a$ = CHR$(0, 72) THEN NN = NN - 5: IF NN < 1 THEN NN = ((NF - 1) ¥ 5
) * 5 + ((NN + 4) MOD 5): IF NN > NF THEN NN = NN - 5
301:   IF a$ = CHR$(0, 75) THEN NN = NN - 1: IF NN < 1 THEN NN = NF
302:   IF a$ = CHR$(0, 80) THEN NN = NN + 5: IF NN > NF THEN NN = ((NN - 6) MO
D 5) + 2: IF NN > 5 THEN NN = NN - 5
303:
304:   CALL PRN.FIL(NO, ATT%(NO), FLN$(NO))
305:   GOTO JP.CUR1
306:
307: JP.CUR2: COLOR 7
308: END SUB
309:
310: DEFINT X-Y
311: DEFSNG T
312: SUB DISP.AXIS
313: '--- 枠の表示 ---
314: '---[BOX]---
315: LINE (0, 0)-(639, 399), 3, B: LINE (0, 127)-(639, 127), 3
316: LINE (79, 127)-(79, 399), 3: LINE (111, 0)-(111, 127), 3
317: LINE (111, 370)-(611, 370), 3
318: LINE (111, 0)-(111, 370), 3: LINE (611, 0)-(611, 370), 3
319: LINE (111, 112)-(611, 112), 3: LINE (111, 210)-(611, 210), 3
320: FOR I% = 1 TO 5: Y% = 210 + (I% - 1) * 40
321:   LINE (111, Y%)-(611, Y%), 3
322: NEXT I%
323: LINE (111, 144)-(611, 144), 3: LINE (111, 176)-(611, 176), 3
324: '---[TITLE]---
325: COLOR 3: ICOLOR% = 3
326: a$ = "走行状態の解析": FOR I = 1 TO 7
327: LOCATE I, 78: PRINT MID$(a$, (I - 1) * 2 + 1, 2); : NEXT I
328: LOCATE 1, 1: PRINT "センサ区間番号": LOCATE 2, 1: PRINT "区 間 長 (km)"

```

```

329: LOCATE 3, 1: PRINT "走行時間 (秒)": LOCATE 4, 1: PRINT "停止回数 "
330: LOCATE 5, 1: PRINT "平均速度(km/h)": LOCATE 6, 1: PRINT "加速度 (m/s2)"
331: LOCATE 7, 1: PRINT "加加速度(m/s3)": LOCATE 8, 1: PRINT "MARKER & JAM"
332: LOCATE 9, 1: PRINT "データ名": LOCATE 11, 1: PRINT "測定年月日"
333: LOCATE 13, 1: PRINT "開始時間 ": LOCATE 15, 1: PRINT "走行時間 "
334: LOCATE 17, 1: PRINT "走行距離 ": LOCATE 21, 1: PRINT "平均速度 "
335: LOCATE 23, 1: PRINT "コメントNO": LOCATE 19, 1: PRINT "距離誤差率"
336: '--[GRAPH]--
337: LOCATE 9, 11: PRINT "v' ": LOCATE 11, 11: PRINT "v'"
338: LOCATE 17, 11: PRINT "速": LOCATE 20, 11: PRINT "度"
339: FOR I = 0 TO 4: N = I * 20: XX = 87: YY = 366 - I * 40: NN$ = STR$(N): NN = L
    EN(NN$) - 1
340: FOR J = 1 TO NN: X = XX + (3 - NN) * 8 + (J - 1) * 8: Y = YY
341: N$ = MID$(NN$, J + 1, 1): CALL PUT.N(X, Y, N$, ICOLOR%): NEXT J
342: NEXT I
343: C$ = "Km/h": FOR I = 1 TO 4: X = I * 8 - 6 + 78: Y = 350: N$ = MID$(C$, I, 1)
    : CALL PUT.N(X, Y, N$, ICOLOR%): NEXT
344: LOCATE 25, 34: PRINT "距": : LOCATE 25, 50: PRINT "離":
345: C$ = "(Km)": FOR I = 1 TO 4: X = I * 8 + 400 + 79: Y = 382: N$ = MID$(C$, I,
    1): CALL PUT.N(X, Y, N$, ICOLOR%): NEXT
346: LOCATE 17, 79: PRINT "時": : LOCATE 20, 79: PRINT "間":
347: C$ = "min": FOR I = 1 TO 3: X = I * 8 + 528 + 79: Y = 350: N$ = MID$(C$, I, 1
    ): CALL PUT.N(X, Y, N$, ICOLOR%): NEXT
348: END SUB
349:
350: DEFDBL T
351: SUB DISP.HEAD
352: '-- ヘッダー部の表示 --
353: COLOR 6: DMAX = 0: X = 1
354: CALL HEAD.READ
355: COEF = ADJ.ROAD(COEF)
356: IF DMAX < TD THEN DMAX = TD
357: LOCATE 10, X: PRINT USING " & &": NA.file$:
358: LOCATE 12, X: PRINT USING " ##/##/##": Year; Month; Day;
359: LOCATE 14, X: PRINT USING " ##:##:##": Shor; Smin; Ssec;
360: LOCATE 16, X: PRINT USING " ##:##:##": t1; t2; T3;
361: LOCATE 18, X: PRINT USING " ##.## km": TD / 1000;
362: LOCATE 20, X: PRINT USING " #.#### ": DEF.DIS;
363: LOCATE 22, X: PRINT USING " ##.##km/h": TD / (t1 + t2 / 60 + T3 / 3600) / 10
    00;
364: LOCATE 24, X: PRINT USING " ####": COMENT;
365: END SUB
366:
367: SUB DISP.RES
368: FOR NN = 1 TO NO.sen
369: Y = KBPOS(NN): IF NO.sen = 1 THEN Y = 45
370: IF Y < 15 OR Y > 75 THEN GOTO DPRES.1
371: SN = INT(SENS.NO(NN) / 100)
372: LOCATE 1, Y: PRINT USING "####": SN;
373: IF ROAD.SW = 1 THEN

```

```

374:          DDT = DIS.ST(NO.st(NN, 2)) - DIS.ST(NO.st(NN, 1))
375:      ELSE
376:          DDT = ALDST
377:      END IF
378:      IF NO.sen = 1 THEN DDT = ALDST
379:      RUNT = T.CON(NN) / 100: AV = 3.6 * DDT / RUNT
380:      LOCATE 2, Y: PRINT USING "##.##"; DDT / 1000;
381:      LOCATE 3, Y: PRINT USING "####"; RUNT;
382:      LOCATE 4, Y: PRINT USING "####"; N.CON(NN);
383:      LOCATE 5, Y: PRINT USING "##.##"; 360 * DDT / T.CON(NN);
384:      LOCATE 6, Y: PRINT USING "##.##"; TOTA(NN) / RUNT;
385:      LOCATE 7, Y: PRINT USING "##.##"; TOTA2(NN) / RUNT;
386: DPRES.1: NEXT NN
387: END SUB
388:
389: DEFSNG T
390: SUB DSP.STAT (KDS, KDE)
391: '-- 交差点名称の表示 --
392:   KII = 12: FOR KDD = KDS TO KDE: LOCATE KII + 1, 71
393:   PRINT USING "##&      &"; KDD; NA.ST$(KDD)
394:   KII = KII + 1: NEXT
395:   LOCATE 24, 71: PRINT "[D]own[U]p";
396: END SUB
397:
398: DEFDBL T
399: SUB HEAD.READ
400: '-- ヘッダ部入力 --
401:
402: OPEN HEAD.fil$ FOR RANDOM AS #1 LEN = 8: a$ = ""
403: FIELD #1, 8 AS B$: FOR I = 1 TO 4: GET #1: a$ = a$ + B$: NEXT
404: ID = ASC(MID$(a$, 2, 1)): COMENT = 10 * (ID ¥ 16) + (ID MOD 16)
405: ID = ASC(MID$(a$, 3, 1)): COMENT = COMENT * 100 + 10 * (ID ¥ 16) + (ID MOD 16
)
406: ' ID = ASC(MID$(a$, 4, 1)): Year = 10 * (ID ¥ 16) + (ID MOD 16): '年
407: ID = ASC(MID$(a$, 5, 1)): Month = 10 * (ID ¥ 16) + (ID MOD 16): '月
408: Year = 91: IF Month < 10 THEN Year = 92
409: ID = ASC(MID$(a$, 6, 1)): Day = 10 * (ID ¥ 16) + (ID MOD 16): '日
410: ID = ASC(MID$(a$, 7, 1)): Shor = 10 * (ID ¥ 16) + (ID MOD 16): '開始時
411: ID = ASC(MID$(a$, 8, 1)): Smin = 10 * (ID ¥ 16) + (ID MOD 16): '開始分
412: ID = ASC(MID$(a$, 9, 1)): Ssec = 10 * (ID ¥ 16) + (ID MOD 16): '開始秒
413: ID = ASC(MID$(a$, 10, 1)): Ehor = 10 * (ID ¥ 16) + (ID MOD 16): '終了時
414: ID = ASC(MID$(a$, 11, 1)): Emin = 10 * (ID ¥ 16) + (ID MOD 16): '終了分
415: ID = ASC(MID$(a$, 12, 1)): Bsec = 10 * (ID ¥ 16) + (ID MOD 16): '終了秒
416: Dstep = ASC(MID$(a$, 13, 1)): 'サンプリング間隔
417: ' STIRE = ASC(MID$(a$, 14, 1)) + 256 * ASC(MID$(a$, 15, 1)): '基準タイ
418: ' ATIRE = ASC(MID$(a$, 16, 1)) + 256 * ASC(MID$(a$, 17, 1)): '補正タイ
419: ' SDIST=ASC(MID$(a$, 18, 1))+256*ASC(MID$(a$, 19, 1)) : '積算距離
420: Stime = ASC(MID$(a$, 22, 1)) + 256! * ASC(MID$(a$, 23, 1)): '積算時間
421: Nstep = ASC(MID$(a$, 24, 1)) + 256! * ASC(MID$(a$, 25, 1)): 'ステップ数
422: COEF = .19623 * Dstep: TD = Nstep * COEF: '1ステップの距離

```

```

423: TT = Stime: t1 = TT ¥ 3600: TT = TT - t1 * 3600: t2 = TT ¥ 60: T3 = TT MOD 60
    : '計測時間
424: FOR I = 1 TO 8: GET #1, I + 4: TIT$(I) = B$: NEXT: 'マ-カ種別
425: CLOSE #1
426: END SUB
427:
428: SUB INIT.CON
429: DIM DTA AS DtaType
430: DIM FF$(20)
431:
432: '-- 走行データ設定 --
433:
434:     SCREEN 88: CLS
435:     NO.file = 1: LINE.SW = 0
436: G1: PRINT
437:     COLOR 6: PRINT "* 走行状態解析プログラム *": PRINT : PRINT
438:     PRINT "走行調査データを装置Aに挿入して下さい"
439:     PRINT "  確認したら[RETURN]を押して下さい  "
440:     WHILE INKEY$ = "": WEND
441:     '    Unit$ = "A:"
442:     a$ = Unit$ + "*.HED": ISW% = CHK.FILE%(a$)
443:     IF NOT ISW% THEN BEEP: PRINT : GOTO G1
444:     CLS : COLOR 3: LOCATE 2, 20: PRINT "調査データの選択";
445:     CALL SELE.FIL(a$, Unit$, FIL$): CLS
446:     LL = LEN(FIL$): a$ = LEFT$(FIL$, LL - 4): NA.file$ = a$
447:     HEAD.fil$ = Unit$ + a$ + ".HED"
448:     STEP.fil$ = Unit$ + a$ + ".STP"
449:     CALL HEAD.READ
450:
451: '-- 路線データ設定 --
452:
453:     CLS : COLOR 6
454:     ROAD.SW = 1
455:     INPUT "路線データを入力しますか(Y/N)"; YN$
456:     IF YN$ = "Y" OR YN$ = "y" THEN GOTO G2
457:     ROAD.SW = 0: NO.sen = 1: W = 1
458:     GOTO G7
459: G2: PRINT
460:     PRINT "路線データを装置Bに挿入して下さい"
461:     PRINT "確認したら[RETURN]を押して下さい  "
462:     WHILE INKEY$ = "": WEND
463:     '    Unitr$ = "B:"
464:     a$ = Unitr$ + "*.ROD": ISW% = CHK.FILE%(a$)
465:     IF NOT ISW% THEN
466:         BEEP: PRINT : COLOR 4: PRINT "路線データがありません!"
467:         PRINT "路線データがない場合、集計データの作成はできません"
468:         INPUT "路線データなしで続けますか(Y/N)"; ISW$
469:         IF ISW$ = "Y" OR ISW$ = "y" THEN
470:             ROAD.SW = 0: NO.sen = 1: W = 1
471:             GOTO G7

```

```

472:             ELSE
473:                 CLS : COLOR 6: GOTO G2
474:             END IF
475:         END IF
476:         CLS : COLOR 3: LOCATE 2, 20: PRINT "路線データの選択";
477:         CALL SELE.FIL(a$, Unitr$, FIL$): CLS
478:         LL = LEN(FIL$): a$ = LEFT$(FIL$, LL - 4)
479:         ROAD.fil$ = Unitr$ + a$ + ".ROD"
480:         CALL ROAD.READ
481:
482: '---集計ファイルの設定---
483:
484:         CLS : COLOR 6: CONG.SW = 1
485:         PRINT "現在の集計データ名": PRINT
486:         FUNC% = 0
487:         NOFIL% = 0
488:         DO WHILE (1)
489:             FPATH$ = Unitr$ + "*.SUM"
490:             FI$ = GetFile$(FPATH$, ATR%, DTA, FUNC%)
491:             IF (FI$ = "") THEN EXIT DO
492:             FUNC% = 1
493:             NOFIL% = NOFIL% + 1
494:             PRINT FI$
495:             K% = 1
496:             DO WHILE K% < 9
497:                 IF MID$(FI$, K%, 1) = "." THEN EXIT DO
498:                 K% = K% + 1
499:             LOOP
500:             FF$(NOFIL%) = LEFT$(FI$, K% - 1)
501:         LOOP
502:         PRINT : PRINT "集計データ名を入力して下さい:"
503:         INPUT "( [RETURN] のみの場合は、集計データを作成しません)"; a$
504:         IF a$ = "" THEN JAM.SW = 0 ELSE JAM.SW = 1
505:         INEW% = 0
506:         FOR I% = 1 TO NOFIL%
507:             IF a$ = FF$(I%) THEN INEW% = 1
508:         NEXT I%: PRINT INEW%
509:         CONG.fil$ = Unitr$ + a$ + ".SUM"
510:         TEMP.fil$ = Unitr$ + a$ + ".TMP"
511:
512: '--- 天候・交通量の入力 ---
513:
514:         CLS : COLOR 6
515:         PRINT USING "対象調査データは、& &"; NA.file$;
516:         PRINT USING " (##月##日##時##分調査開始"; Month; Day; Shor; Smin;
517:         PRINT USING " コメント ####) です."; COMENT: PRINT
518:
519: G3: PRINT "調査時の天候 (1:晴 2:曇 3:霧 4:小雨 5:雨 6:雪) ";
520:         INPUT N: IF N < 0 OR N > 6 THEN BEEP: GOTO G3 ELSE WHETHER(1) = N
521: G4: PRINT "調査時の視界 (1:良好 2:夕暮れ 3:暗がり) ";

```

```

522:      INPUT N: IF N < 0 OR N > 3 THEN BEEP: GOTO G4 ELSE WHETHER(2) = N
523: G5: PRINT "調査時の路面状況 (1:乾燥 2:湿潤 3:新雪 4:圧雪 5:凍結)
";
524:      INPUT N: IF N < 0 OR N > 5 THEN BEEP: GOTO G5 ELSE WHETHER(3) = N
525: G6: PRINT "当該方向の車種別時間交通量(台/時)を入力して下さい"
526:      INPUT "全車種合計"; TRAFFIC(1)
527:      INPUT "大型車合計"; TRAFFIC(2)
528:      INPUT "よろしいですか (Y/N) "; Y$
529:      IF Y$ <> "y" AND Y$ <> "Y" THEN GOTO G3
530: G7: CLS
531: END SUB
532:
533: SUB MK.FIL (NPSC)
534: '-- 渋滞FILE作成 --
535: DIM LLMAXP(30), LLMAX(30), CHK$(30)
536:
537: IF INEW% <> 0 THEN
538:     OPEN TEMP.fil$ FOR OUTPUT AS #3
539: ELSE
540:     OPEN CONG.fil$ FOR OUTPUT AS #3
541: END IF
542:
543: IF Smin > 50 THEN Shor = Shor + 1
544:
545: FOR NN% = 1 TO NO.sen
546:     PRINT #3, USING "#####"; CITY.NO(NN%);
547:     PRINT #3, USING "#####"; SENS.NO(NN%);
548:     PRINT #3, USING "#####"; ROAD.NO(NN%);
549:     PRINT #3, USING "#"; CLASS%(NN%);
550:     DDT = DIS.ST(NO.st(NN%, 2)) - DIS.ST(NO.st(NN%, 1))
551:     PRINT #3, USING "###"; DDT / 100;
552:     PRINT #3, USING "#"; HOUKOU(NN%);
553:     PRINT #3, USING "#"; WN(NN%);
554:
555:     PRINT #3, USING "##"; Year; Month; Day; Shor;
556:     PRINT #3, USING "#"; WHETHER(1); WHETHER(2); WHETHER(3);
557:     PRINT #3, USING "##"; FHOR(NN%, 1); FMIN(NN%, 1); FSEC(NN%, 1);
558:     PRINT #3, USING "##"; FHOR(NN%, 2); FMIN(NN%, 2); FSEC(NN%, 2);
559:     PRINT #3, USING "#####"; TRAFFIC(1); TRAFFIC(2);
560:     TS = T.CON(NN%) / 100: TH = INT(TS / 3600): TS = TS - TH * 3600
561:     TM = INT(TS / 60): TS = TS - TM * 60
562:     PRINT #3, USING "##"; TH; TM; TS;
563:     PRINT #3, USING "###"; N.CON(NN%);
564:     PRINT #3, USING "#####"; 3600 * DDT / T.CON(NN%);
565:     RUNT = T.CON(NN%) / 100
566:     PRINT #3, USING "#####"; TOTA(NN%) / RUNT * 100;
567:     PRINT #3, USING "#####"; TOTA2(NN%) / RUNT * 100;
568:
569:     PRINT #3, USING "#####"; S.CON(NN%) / 100;
570:

```

```

571:   FOR I% = 0 TO 30: LLMAX(I%) = 0: LLMAXP(I%) = 0: NEXT I%
572:   LP% = 0
573:   LL1 = 0: LL2 = 0: LLTIM1 = 0: LLTIM2 = 0
574:   DD1 = DIS.ST(NO.st(NN%, 1)): DD2 = DIS.ST(NO.st(NN%, 2))
575:   TT1 = SET.TIM(NN%, 1): TT2 = SET.TIM(NN%, 2)
576:
577:   FOR I = 1 TO NPSC
578:     IF PC(I) <> 0 THEN
579:       P.L1 = COEF * PS(I, 1): P.L2 = COEF * PS(I, 2)
580:       P.T1 = P.TIM(I, 1): P.T2 = P.TIM(I, 2)
581:       IF P.L2 > DD1 AND P.L1 < DD2 THEN
582:         IF P.L1 < DD1 THEN P.L1 = DD1: P.T1 = TT1
583:         IF P.L2 > DD2 THEN P.L2 = DD2: P.T2 = TT2
584:         DDT = P.L2 - P.L1: DDTIM = P.T2 - P.T1
585:         IF PC(I) = 1 THEN
586:           LL1 = LL1 + DDT: LLTIM1 = LLTIM1 + DDTIM
587:         ELSE
588:           LL2 = LL2 + DDT: LLTIM2 = LLTIM2 + DDTIM
589:         END IF
590:         DCHK = ABS(LLMAXP(LP%) + LLMAX(LP%) + DD1 - P.L1)
591:         IF DCHK < 5 THEN
592:           LLMAX(LP%) = LLMAX(LP%) + DDT
593:         ELSE
594:           LP% = LP% + 1
595:           LLMAX(LP%) = DDT: LLMAXP(LP%) = P.L1 - DD1
596:         END IF
597:       END IF
598:     END IF
599:   NEXT I
600:
601:   IF LP% > 1 THEN
602:     FOR I = 1 TO LP%: FOR J = I TO LP%
603:       IF LLMAX(I) < LLMAX(J) THEN
604:         SWAP LLMAX(I), LLMAX(J): SWAP LLMAXP(I), LLMAXP
605:       END IF
606:     NEXT J, I
607:   END IF
608:
609:   PRINT #3, USING "####": LL1 / 10;
610:   PRINT #3, USING "####": LL2 / 10;
611:   PRINT #3, USING "####": LLTIM1;
612:   PRINT #3, USING "####": LLTIM2;
613:   LLTIMT = LLTIM1 + LLTIM2
614:   IF LLTIMT > .01 THEN
615:     PRINT #3, USING "####": 36 * (LL1 + LL2) / LLTIMT;
616:   ELSE
617:     PRINT #3, "  0";
618:   END IF
619:   FOR I = 1 TO 3

```

```

620:          PRINT #3, USING "###"; LLMAXP(1) / 100;
621:          PRINT #3, USING "###"; LLMAX(1) / 100;
622:      NEXT I
623:      PRINT #3, USING "#"; F.CON(NN%)
624:
625:  NEXT NN%
626:  CLOSE #3
627:  IF INEW% <> 0 THEN
628:      OPEN TEMP.fil$ FOR INPUT AS #3
629:      N% = 0
630:      WHILE NOT EOF(3)
631:          LINE INPUT #3, IN$
632:          N% = N% + 1
633:          CHK$(N%) = LEFT$(IN$, 29)
634:      WEND
635:      CLOSE #3
636:      OPEN CONG.fil$ FOR INPUT AS #3
637:      OPEN TEMP.fil$ FOR APPEND AS #4
638:      WHILE NOT EOF(3)
639:          LINE INPUT #3, IN$
640:          OUT$ = LEFT$(IN$, 29)
641:          ISW% = 0
642:          FOR I% = 1 TO N%
643:              IF (OUT$ = CHK$(I%)) THEN ISW% = 1
644:          NEXT I%
645:          IF ISW% = 0 THEN PRINT #4, IN$
646:      WEND
647:      CLOSE #3: CLOSE #4
648:      KILL CONG.fil$
649:      NAME TEMP.fil$ AS CONG.fil$
650:  END IF
651:  END SUB
652:
653:  DEFSNG T, X-Y
654:  SUB MOV.DAT (II, NPSC)
655:      PS(II, 2) = PS(II + 1, 2)
656:      P.TIM(II, 2) = P.TIM(II + 1, 2)
657:      P.STP(II) = P.STP(II) + P.STP(II + 1)
658:      FOR KK = 1 TO 8
659:          P.MRK(II, KK) = P.MRK(II, KK) + P.MRK(II + 1, KK)
660:      NEXT
661:      IF P.MAX(II) < P.MAX(II + 1) THEN P.MAX(II) = P.MAX(II + 1)
662:      FOR J = II + 2 TO NPSC
663:          PC(J - 1) = PC(J)
664:          PS(J - 1, 1) = PS(J, 1): PS(J - 1, 2) = PS(J, 2)
665:          P.TIM(J - 1, 1) = P.TIM(J, 1): P.TIM(J - 1, 2) = P.TIM(J, 2)
666:          P.MAX(J - 1) = P.MAX(J)
667:          P.STP(J - 1) = P.STP(J)
668:          FOR KK = 1 TO 8
669:              P.MRK(J - 1, KK) = P.MRK(J, KK)

```

```

670:             NEXT KK
671:     NEXT J
672:     NPSC = NPSC - 1
673: END SUB
674:
675: SUB PRN.FIL (N, ATT%, FLN$)
676:   NN = ABS(N): IF N < 0 THEN COLOR 11 + ATT% ELSE COLOR 3 + ATT%
677:   LY = ((NN - 1) MOD 5) * 15 + 2: LX = (NN - 1) ¥ 5 + 1 + 5
678:   LOCATE LX, LY: IF ATT% = 0 THEN PRINT FLN$: ELSE PRINT "<"; FLN$: ">";
679: END SUB
680:
681: SUB RE.CHK (LDIS, NPSC)
682: '--- 渋滞区間の補正 ---
683: GF1: IF NPSC <= 2 THEN GOTO GFO
684:     FOR I = 2 TO NPSC - 1: IF PC(I) <> 0 THEN GOTO GF2
685:         C1 = (PS(I, 2) - PS(I, 1)) * COEF: IF C1 > LDIS THEN GOTO GF2
686:         CALL RE.CHK2(I, LDIS, NPSC)
687:         GOTO GF1
688: GF2: NEXT I
689: GF3: IF NPSC <= 2 THEN GOTO GFO
690:     FOR I = 2 TO NPSC - 1: IF PC(I) = 0 THEN GOTO GF5
691:         FOR M = 1 TO NPSC: IF PC(M) = 0 THEN GOTO GF4
692:         NEXT M
693: GF4:   C1 = (PS(M - 1, 2) - PS(I, 1)) * COEF
694:         IF C1 > LDIS THEN I = M - 1: GOTO GF5
695:         CALL RE.CHK2(M - 1, LDIS, NPSC)
696:         GOTO GF3
697: GF5: NEXT I
698: GF6: IF NPSC <= 2 THEN GOTO GFO
699:     FOR I = 2 TO NPSC - 1: IF PC(I) <> 1 THEN GOTO GF7
700:         IF PC(I - 1) = 0 OR PC(I + 1) = 0 THEN GOTO GF7
701:         C1 = (PS(I, 2) - PS(I, 1)) * COEF: IF C1 > LDIS THEN GOTO GF7
702:         CALL RE.CHK2(I, LDIS, NPSC)
703:         GOTO GF6
704: GF7: NEXT I
705: GF8: IF NPSC <= 2 THEN GOTO GFO
706:     FOR I = 2 TO NPSC - 1: IF PC(I) <> 2 THEN GOTO GF9
707:         IF PC(I - 1) = 0 OR PC(I + 1) = 0 THEN GOTO GF9
708:         C1 = (PS(I, 2) - PS(I, 1)) * COEF: IF C1 > LDIS THEN GOTO GF9
709:         CALL RE.CHK2(I, LDIS, NPSC)
710:         GOTO GF8
711: GF9: NEXT I
712: GFO:
713: END SUB
714:
715: DEFDBL T
716: SUB RE.CHK2 (I, LDIS, NPSC)
717: '--- 渋滞間隙による補正 ---
718: DISA = (PS(I - 1, 2) - PS(I - 1, 1)) * COEF
719: DISB = (PS(I + 1, 2) - PS(I + 1, 1)) * COEF

```

```

720: KTA = PC(I - 1): KTB = PC(I + 1)
721: IF KTA = KTB THEN
722:     PC(I - 1) = KTA
723:     II = I - 1: CALL MOV.DAT(II, NPSC): CALL MOV.DAT(II, NPSC)
724: ELSE
725:     IF DISA < DISB THEN
726:         PC(I) = KTB
727:         II = I: CALL MOV.DAT(II, NPSC)
728:     ELSE
729:         PC(I) = KTA
730:         II = I - 1: CALL MOV.DAT(II, NPSC)
731:     END IF
732: END IF
733: II = 1
734: DO WHILE (II <= NPSC - 1)
735:     IF PC(II) = PC(II + 1) THEN
736:         CALL MOV.DAT(II, NPSC)
737:         II = 0
738:     END IF
739:     II = II + 1
740: LOOP
741: END SUB
742:
743: DEFINT X-Y
744: DEFSNG T
745: SUB ROAD.READ
746:     K = 1: KK = 0
747:     OPEN ROAD.fil$ FOR INPUT AS #1
748:     INPUT #1, NO.sen
749:     FOR I = 1 TO NO.sen
750:         LINE INPUT #1, a$
751:         CITY.NO(I) = VAL(LEFT$(a$, 5)): SENS.NO(I) = VAL(MID$(a$, 6, 6))
752:         ROAD.NO(I) = VAL(MID$(a$, 12, 4)): HOUKOU(I) = VAL(MID$(a$, 16, 1))
753:         W = VAL(MID$(a$, 17, 1)): WW(I) = 1: IF W > 2 THEN WW(I) = 2
754:         CLASS%(I) = VAL(MID$(a$, 20, 1))
755:         NN = VAL(MID$(a$, 18, 2)): K1 = K + NN - 1
756:         NO.st(I, 1) = K: NO.st(I, 2) = K1: K = K1
757:         FOR J = 1 TO NN
758:             LINE INPUT #1, a$
759:             D = VAL(MID$(a$, 6, 5)): NA$ = MID$(a$, 11, 22)
760:             IF I <> 1 AND J = 1 THEN GOTO GA1
761:             KK = KK + 1: NA.ST$(KK) = NA$: DIS.ST(KK) = D * 10
762: GA1: NEXT J
763:     NEXT I
764:     NO.stt = KK
765:     FOR I = 1 TO NO.sen
766:         KBOD(I) = DIS.ST(NO.st(I, 2))
767:     NEXT
768:     CLOSE #1
769: END SUB

```

```

770:
771: DEFSNG X-Y
772: DEFDBL T
773: SUB SEL.FIL (PATH$, NF, FLN$(), ATT%())
774: DIM DtaData AS DtaType
775: Wild$ = PATH$: FileName$ = GetFile$(Wild$, Attr%, DtaData, 0)
776: NF = 0
777: DO
778:     NF = NF + 1: FLN$(NF) = FileName$
779:     ATT%(NF) = 0: IF Attr% = &H10 THEN ATT%(NF) = 1
780:     FileName$ = GetFile$(Wild$, Attr%, DtaData, 1)
781: LOOP UNTIL FileName$ = ""
782: 'FOR I = 1 TO NF - 1: FLN$(I) = FLN$(I + 1): ATT%(I) = ATT%(I + 1): NEXT: NF =
    NF - 1
783: END SUB
784:
785: SUB SELE.FIL (a$, U$, f$)
786: REDIM FLN$(100), ATT%(100): COLOR 7
787: PATH$ = a$: CALL SEL.FIL(PATH$, NF, FLN$(), ATT%())
788:
789: JPF1:
790:     LOCATE 4, 5: COLOR 3: PRINT "SELSCT FILE NAME (";
791:     COLOR 4: PRINT "<SUB-DIR>"; : COLOR 3: PRINT " , FILE ,";
792:     COLOR 6: PRINT " [ESC] QUIT , [RETURN] SELECT )"
793:     dx = 75 * 8 + 8: DY = ((NF - 1) ¥ 5 + 1) * 16 + 8
794:     LINE (4, 76)-(dx, DY + 76), 3, B
795:     FOR I = 1 TO NF: CALL PRN.FIL(I, ATT%(I), FLN$(I)): NEXT: PRINT
796:     CALL CUR(NF, FLN$(), ATT%(), NN)
797:
798: 'JPF2: FOR I = LEN(PATH$) - 1 TO 1 STEP -1
799: '     IF MID$(PATH$, I, 1) = "¥" THEN GOTO JPF3
800: '     NEXT I
801: '
802: 'JPF3: U$ = LEFT$(PATH$, I): F$ = MID$(PATH$, I + 1, LEN(PATH$) - I - 1)
803:
804: f$ = FLN$(NN)
805:
806: END SUB
807:
808: DEFINT X-Y
809: SUB SET.TIME
810: '--- 時間軸の設定 ---
811:     TBASE = 0: TMAX = 0
812:     CALL HEAD.READ
813:     IF ROAD.SW = 1 THEN COEF = ADJ.ROAD(COEF)
814:     CALL STEP.OPEN
815:     NST = INT(Mstart / COEF) + 1: NED = INT(Mend / COEF + .05)
816:     IF NED > Nstep THEN NED = Nstep
817:     CALL STEP.READ(NST): TH = SH: TM = SM: TS = SS
818:     CALL STEP.READ(NED)

```

```

819:      TX = (SH - TH) * 3600 + (SM - TM) * 60 + (SS - TS): T = INT(TX / 60 +
1): T = INT(T / 10 + 1) * 10
820:      IF TBASE < T * 60 THEN TBASE = T * 60: TMAX = T
821:      TT = TX: t1 = TT ¥ 3600: TT = TT - t1 * 3600: t2 = TT ¥ 60: T3 = TT MO
D 60: '計測時間
822:      CLOSE #2
823:      ICOLOR% = 3: X = 615: Y = 366: N$ = "0": CALL PUT.N(X, Y, N$, ICOLOR%)
824:      C$ = STR$(TMAX): LL = LEN(C$)
825:      FOR I = 2 TO LL: X = I * 7 + 601: Y = 206: N$ = MID$(C$, I, 1): CALL PUT.N(
X, Y, N$, ICOLOR%): NEXT
826:      C$ = STR$(INT(TMAX * .5 + .5)): LL = LEN(C$)
827:      FOR I = 2 TO LL: X = I * 7 + 601: Y = 286: N$ = MID$(C$, I, 1): CALL PUT.N(
X, Y, N$, ICOLOR%): NEXT
828:
829: ' -- 距離軸の設定 --
830:
831:      DSTART = INT(Mstart / 100 + .5) / 10: DEND = INT(Mend / 100 + .5) / 10
832:      DMID = INT((Mstart + Mend) / 200 + .5) / 10
833:      C$ = STR$(DSTART): LL = LEN(C$)
834:      FOR I = 2 TO LL: X = I * 7 + 10 + 79: Y = 374: N$ = MID$(C$, I, 1): CALL PU
T.N(X, Y, N$, ICOLOR%): NEXT
835:      C$ = STR$(DMID): LL = LEN(C$)
836:      FOR I = 2 TO LL: X = I * 7 + 256 + 79: Y = 374: N$ = MID$(C$, I, 1): CALL P
UT.N(X, Y, N$, ICOLOR%): NEXT
837:      C$ = STR$(DEND): LL = LEN(C$)
838:      FOR I = 2 TO LL: X = I * 7 + 500 + 79: Y = 374: N$ = MID$(C$, I, 1): CALL P
UT.N(X, Y, N$, ICOLOR%): NEXT
839:      IF ROAD.SW = 0 THEN GOTO GB1
840:      FOR I = 1 TO NO.stt
841:          IF DIS.ST(I) < DSTART * 1000 OR DIS.ST(I) > DEND * 1000 THEN GOTO GB2
842:          X = (DIS.ST(I) - DSTART * 1000) * .5 / (DEND - DSTART) + 111
843:          LINE (X, 371)-(X, 372), 7
844:      GB2: NEXT
845:      GB1: ALDST = Mend - Mstart
846:
847:      IF ROAD.SW = 0 THEN
848:          KBPOS(1) = 360 / 8
849:      ELSE
850:          FOR I = 1 TO NO.sen
851:              XS = 111: XE = 601
852:              SS = DIS.ST(NO.st(I, 1)): EE = DIS.ST(NO.st(I, 2))
853:              IF SS > DEND * 1000 OR EE < DSTART * 1000 THEN
854:                  KBPOS(I) = -1
855:              ELSE
856:                  IF SS >= DSTART * 1000 THEN
857:                      XS = (SS - DSTART * 1000) * .5 / (DEND - DSTART) + 111
858:                      LINE (XS, 370)-(XS, 0), 3
859:                  END IF
860:                  IF EE <= DEND * 1000 THEN
861:                      XE = (EE - DSTART * 1000) * .5 / (DEND - DSTART) + 111

```

```

862:             IF I <> NO.sen THEN LINE (XE, 370)-(XE, 0), 3
863:             END IF
864:             KBPOS(I) = (XE + XS) / 2 / 8 - 2
865:         END IF
866:     NEXT
867: END IF
868: END SUB
869:
870: DEFSNG I-K, M-N, T, X-Y
871: SUB SET.W
872:     L.V1 = LIMIT(1, W): L.VM1 = LIMIT(2, W)
873:     L.V2 = LIMIT(3, W): L.VM2 = LIMIT(4, W)
874: END SUB
875:
876: DEFSNG L
877: SUB STEP.OPEN
878: '-- ステップデータOPEN --
879: OPEN STEP.fil$ FOR RANDOM AS #2 LEN = 8: FIELD #2, 8 AS AA$
880: END SUB
881:
882: DEFLNG I-N
883: DEFDBL T
884: SUB STEP.READ (II)
885: '-- ステップ部入力 --
886: GET #2, II
887: 'DIS=ASC(MID$(AA$, 1, 1))+256*ASC(MID$(AA$, 2, 1)) : '走行距離
888: TIM = ASC(MID$(AA$, 3, 1)) + 256! * ASC(MID$(AA$, 4, 1)): '所要時間
889: ID = ASC(MID$(AA$, 5, 1)): SH = 10 * (ID ¥ 16) + (ID MOD 16): '時刻 (時)
890: ID = ASC(MID$(AA$, 6, 1)): SM = 10 * (ID ¥ 16) + (ID MOD 16): '時刻 (分)
891: ID = ASC(MID$(AA$, 7, 1)): SS = 10 * (ID ¥ 16) + (ID MOD 16): '時刻 (秒)
892: ITT = ASC(MID$(AA$, 8, 1))
893: FOR I = 1 TO 8: MK(I) = ITT MOD 2: ITT = ITT ¥ 2: NEXT: 'マ-カ信号
894: END SUB
895:

```

JAMT. BAS

```

1: DECLARE SUB DSORT (NDATA!)
2: DECLARE SUB DWGRAPH1 (TILE$(), NDATA!, CNSNO%, RTNO%, JAM AS ANY)
3: DECLARE SUB DWGRAPH2 (TILE$(), NDATA!, CNSNO%, RTNO%, SMAXMIN!(), JMAXMIN!())
4: DECLARE SUB DWGRAPH3 (NDATA!, CNSNO%, RTNO%, SPDSUM!())
5: DECLARE SUB GRP1 (TILE$(), CNSNO%, RTNO%, YY%, MM%, DD%, WW%, SPD!(), JMD!(),
  CNT%())
6: DECLARE SUB DISP.AXIS1 (TILE$())
7: DECLARE SUB DISP.AXIS2 (TILE$())
8: DECLARE SUB DISP.AXIS3 ()
9: DECLARE SUB DISP.AXIS4 (TILE$())
10: DECLARE SUB Sumdata (ID%, UPDOWN%, TIMIDX%, SPEED%, LTOT!)
11: DECLARE SUB CenseSel (EXEC$, RTNO%, CNSNO%)
12: DECLARE SUB ReadJamf (NDATA!, RTNO%, CNSNO%, JAM AS ANY)
13: DECLARE SUB PR.Outline (NDATA!, JAM AS ANY)
14: DECLARE SUB BARA (IN$, JAM AS ANY)
15: DECLARE SUB IntInput (IX%, IY%, PRMP$, ILN%, IWAIT%, IN$, IN$)
16: DECLARE SUB CloseJamf ()
17: DECLARE SUB OpenJamf (UNIT$, JamF$)
18: DECLARE FUNCTION LeapYear% (YY%)
19: DECLARE FUNCTION DateWeek% (DT AS ANY)
20: DECLARE FUNCTION Holiday% (MM%, DD%)
21: DECLARE SUB LineEdit (IX%, IY%, PRMP$, ILN%, INEW%, IWAIT%, IN$)
22: DECLARE SUB DWGRAPH4 (TILE$(), NDATA!, CNSNO%, RTNO%, JAM AS ANY)
23: DECLARE SUB GRP3 (NDATA!)
24: DECLARE SUB GRP4 (TILE$(), UNI!, CNSNO%, RTNO%, YY%, MM%, DD%, WW%, JPOS!(), J
  LEN!(), CNT%())
25: DECLARE SUB PR.oneline (IN$)
26: DECLARE SUB PR.Sumline (SDATA!(), ND!())
27: DECLARE SUB Sumdata2 (IN$, SD!(), ND!(), JAM AS ANY)
28: '*****
29: '*   [常時観測データ集計プログラム]   *
30: '*****
31: '
32: '$INCLUDE: 'YOSHIDA.BI'
33: '$INCLUDE: 'JAMFN.TYP'
34: '
35: '*****
36: '*   メイン   *
37: '*****
38: '
39: DIM JAM AS JamF
40: DIM DTA AS DtaType
41: DIM SHARED SPDDAT!(13, 2, 24), DTCNT%(13, 2, 24)
42: DIM SHARED JAMDAT!(13, 2, 24), SPDSUM!(2, 70)
43: DIM SHARED SMAXMIN!(13, 2, 3), JMAXMIN!(13, 2, 3)
44: DIM SHARED INDT(350) AS STRING * 115
45: DIM TILE$(2)
46: COMMON SHARED UNIT$, COPY.SW%

```

```

47: KEY 15, CHR$(&H0) + CHR$(&H60)
48: KEY 16, CHR$(&H2) + CHR$(&H60)
49: KEY(16) ON
50: KEY(15) ON
51: ON KEY(15) GOSUB key.stop
52: ON KEY(16) GOSUB key.stop
53: SCREEN 88: CLS : COLOR 3
54: PRINT "作業内容を入力して下さい"
55: INPUT "1:自動集計 2:手動表示"; COPY.SW%
56: IF COPY.SW% < 1 OR COPY.SW% > 2 THEN COPY.SW% = 1
57:
58: IF COPY.SW% = 1 THEN OPEN "PRN" FOR OUTPUT AS #2
59:
60: TILE$(1) = "": TILE$(2) = ""
61: A$ = CHR$(VAL("&H44")) + CHR$(0) + CHR$(VAL("&H44")) + CHR$(0)
62: B$ = CHR$(0) + CHR$(0) + CHR$(0) + CHR$(0)
63: C$ = CHR$(VAL("&H11")) + CHR$(0) + CHR$(VAL("&H11")) + CHR$(0)
64: FOR J = 1 TO 2: TILE$(2) = TILE$(2) + A$ + B$ + C$ + B$: NEXT
65: A$ = CHR$(VAL("&H55")) + CHR$(0) + CHR$(VAL("&H55")) + CHR$(0)
66: B$ = CHR$(VAL("&HAA")) + CHR$(0) + CHR$(VAL("&HAA")) + CHR$(0)
67: FOR J = 1 TO 4: TILE$(1) = TILE$(1) + A$ + B$: NEXT
68:
69: '---ドライブ選択---
70:
71: UNIT$ = "B:"
72:
73: DO WHILE (1)
74:     CLS
75:     CALL CenseSel("渋滞情報データの集計", RTNO%, CNSNO%)
76:     IF (RTNO% = 0) THEN EXIT DO
77:     CLS
78:     LOCATE 10, 32: PRINT "渋滞情報集計中"
79:     FOR I% = 1 TO 24
80:         FOR J% = 1 TO 13
81:             FOR K% = 1 TO 2
82:                 SPDDAT!(J%, K%, I%) = 0!
83:                 DTCNT%(J%, K%, I%) = 0
84:                 JAMDAT!(J%, K%, I%) = 0!
85:             NEXT K%
86:         NEXT J%
87:     NEXT I%
88:     FOR I% = 1 TO 70: FOR J% = 1 TO 2: SPDSUM!(J%, I%) = 0: NEXT J%
89:     , I%
90: '---渋滞ファイルの入力---
91: CALL ReadJamf(NDATA, RTNO%, CNSNO%, JAM)
92:
93: CALL DSORT(NDATA)
94:
95: '---集計---

```

```

96: FOR I% = 1 TO NDATA
97:     CALL BARA(INDT$(I%), JAM)
98:
99:     IF (I% = 1) THEN ALDIST! = VAL(JAM.ALLWAY) / 10!
100:    TIMIDX% = VAL(JAM.DATE.HH)
101:    IF (TIMIDX% >= 24) THEN TIMIDX% = 0
102:    TIMIDX% = TIMIDX% + 1
103:
104:    UPDWN% = VAL(JAM.DIRECT)
105:    IF (UPDWN% <> 1 AND UPDWN% <> 2) THEN UPDWN% = 1
106:
107:    TYP1% = DateWeek%(JAM.DATE): TYP% = (TYP1% MOD 7) + 1
108:    '---平成4年度分の日付分類---
109:    YY% = VAL(JAM.DATE.YEAR): MM% = VAL(JAM.DATE.MON)
110:    IF YY% > 91 THEN
111:        TYP1% = 8
112:        IF MM% > 3 THEN TYP1% = TYP1% + 2
113:        IF MM% > 6 THEN TYP1% = TYP1% + 2
114:        IF TYP% <> 1 THEN TYP1% = TYP1% + 1
115:        TYP% = TYP1%
116:    END IF
117:    '---平成4年度分の日付分類---
118:
119:    SPEED% = VAL(JAM.SPEED) / 10!
120:    LTOT! = VAL(JAM.JAMLEN1) * 10! + VAL(JAM.JAMLEN2) * 10!
121:    KS% = SPEED%: IF KS% > 60 THEN KS% = 60
122:    IF KS% <= 0 THEN KS% = 1
123:    SPDSUM!(UPDWN%, KS%) = SPDSUM!(UPDWN%, KS%) + 1
124:    CALL Sumdata(TYP%, UPDWN%, TIMIDX%, SPEED%, LTOT!)
125: NEXT I%
126:
127: '---平均値の算定---
128:     FOR J% = 1 TO 2
129:         FOR K% = 1 TO 13
130:             SNN% = 0: SPDMAX! = 0: SPDMIN! = 9999: SPDAVG! = 0
131:             JNN% = 0: JAMMAX! = 0: JAMMIN! = 9999: JAMAVG! = 0
132:             FOR I% = 1 TO 24
133:                 IF (DTCNT%(K%, J%, I%) > 0) THEN
134:                     CNT! = CSNG(DTCNT%(K%, J%, I%))
135:                     SPD! = SPDDAT!(K%, J%, I%) / CNT!
136:                     JLN! = JAMDAT!(K%, J%, I%) / CNT!
137:                     SPDDAT!(K%, J%, I%) = SPD!
138:                     JAMDAT!(K%, J%, I%) = JLN!
139:                     SNN% = SNN% + 1
140:                     SPDAVG! = SPDAVG! + SPD!
141:                     IF (SPDMAX! < SPD!) THEN SPDMAX! = SPD!
142:                     IF (SPDMIN! > SPD!) THEN SPDMIN! = SPD!
143:                     IF JLN! > 0 THEN
144:                         JNN% = JNN% + 1
145:                         JAMAVG! = JAMAVG! + JLN!

```

```

146:                                     IF (JAMMAX! < JLN!) THEN JAMMAX
! = JLN!
147:                                     IF (JAMMIN! > JLN!) THEN JAMMIN
! = JLN!
148:                                     END IF
149:                                 END IF
150:                            NEXT I%
151:                            IF SNN% <> 0 THEN SPDAVG! = SPDAVG! / SNN%
152:                            IF JNN% <> 0 THEN JMAVG! = JMAVG! / JNN%
153:                            SMAXMIN!(K%, J%, 1) = SPDMAX!
154:                            SMAXMIN!(K%, J%, 2) = SPDMIN!
155:                            SMAXMIN!(K%, J%, 3) = SPDAVG!
156:                            JMAXMIN!(K%, J%, 1) = JAMMAX!
157:                            JMAXMIN!(K%, J%, 2) = JAMMIN!
158:                            JMAXMIN!(K%, J%, 3) = JMAVG!
159:                        NEXT K%
160:                NEXT J%
161:
162:                FOR J% = 1 TO 2
163:                    N! = 0
164:                    FOR I% = 1 TO 60
165:                        N! = N! + SPDSUM!(J%, I%)
166:                    NEXT I%
167:                    T! = 0
168:                    FOR I% = 1 TO 60
169:                        T! = T! + SPDSUM!(J%, I%)
170:                        IF N! <> 0 THEN SPDSUM!(J%, I%) = 100! * T! / N!
171:                    NEXT I%
172:                NEXT J%
173:
174:                '----グラフ作成----
175:                CALL DWGRAPH2(TILE$, NDATA, CNSNO%, RTNO%, SMAXMIN!(), JMAXMIN!())
176:                CALL DWGRAPH3(NDATA, CNSNO%, RTNO%, SPDSUM!())
177:                CALL DWGRAPH1(TILE$, NDATA, CNSNO%, RTNO%, JAM)
178:                CALL DWGRAPH4(TILE$, NDATA, CNSNO%, RTNO%, JAM)
179:                '----概要印字----
180:                CALL PR.Outline(NDATA, JAM)
181:
182:                EXIT DO
183:            LOOP
184:            CLS
185:            END
186:
187:            HolidayData:
188:            DATA 2, 1, 15
189:            DATA 1, 11
190:            DATA 1, 21
191:            DATA 1, 29
192:            DATA 3, 3, 4, 5
193:            DATA 0

```

```

194: DATA      0
195: DATA      0
196: DATA      2,    15,23
197: DATA      1,    10
198: DATA      2,    3,23
199: DATA      1,    23
200:
201: key.stop:
202:          CLOSE : CLS : END
203: RETURN
204:
205: SUB BARA (IN$, JAM AS JamF)
206:          JAM.CODE.SHIRE = MID$(IN$, 1, 2)
207:          JAM.CODE.CITY = MID$(IN$, 3, 3)
208:          JAM.CODE.CENSNO = MID$(IN$, 6, 4)
209:          JAM.CODE.PARTNO = MID$(IN$, 10, 2)
210:          JAM.RTNO = MID$(IN$, 12, 4)
211:          JAM.CLASS = MID$(IN$, 16, 1)
212:          JAM.ALLWAY = MID$(IN$, 17, 3)
213:          JAM.DIRECT = MID$(IN$, 20, 1)
214:          JAM.WW = MID$(IN$, 21, 1)
215:          JAM.DATE.YEAR = MID$(IN$, 22, 2)
216:          JAM.DATE.MON = MID$(IN$, 24, 2)
217:          JAM.DATE.DAY = MID$(IN$, 26, 2)
218:          JAM.DATE.HH = MID$(IN$, 28, 2)
219:          JAM.WHETER1 = MID$(IN$, 30, 1)
220:          JAM.WHETER2 = MID$(IN$, 31, 1)
221:          JAM.WHETER3 = MID$(IN$, 32, 1)
222:          JAM.TSTART.HH = MID$(IN$, 33, 2)
223:          JAM.TSTART.MM = MID$(IN$, 35, 2)
224:          JAM.TSTART.SS = MID$(IN$, 37, 2)
225:          JAM.TEND.HH = MID$(IN$, 39, 2)
226:          JAM.TEND.MM = MID$(IN$, 41, 2)
227:          JAM.TEND.SS = MID$(IN$, 43, 2)
228:          JAM.TRAF.ALL = MID$(IN$, 45, 4)
229:          JAM.TRAF.BIG = MID$(IN$, 49, 4)
230:          JAM.TRAVEL.HH = MID$(IN$, 53, 2)
231:          JAM.TRAVEL.MM = MID$(IN$, 55, 2)
232:          JAM.TRAVEL.SS = MID$(IN$, 57, 2)
233:          JAM.STOPCNT = MID$(IN$, 59, 3)
234:          JAM.SPEED = MID$(IN$, 62, 4)
235:          JAM.ACCEL = MID$(IN$, 66, 4)
236:          JAM.AACCEL = MID$(IN$, 70, 4)
237:          JAM.STOPTIM = MID$(IN$, 74, 4)
238:          JAM.JAMLEN1 = MID$(IN$, 78, 4)
239:          JAM.JAMLEN2 = MID$(IN$, 82, 4)
240:          JAM.JAMTIM1 = MID$(IN$, 86, 4)
241:          JAM.JAMTIM2 = MID$(IN$, 90, 4)
242:          JAM.JAMSPED = MID$(IN$, 94, 4)
243:          JAM.JAM1.PS = MID$(IN$, 98, 3)

```

```

244:      JAM. JAM1. LN = MID$(IN$, 101, 3)
245:      JAM. JAM2. PS = MID$(IN$, 104, 3)
246:      JAM. JAM2. LN = MID$(IN$, 107, 3)
247:      JAM. JAM3. PS = MID$(IN$, 110, 3)
248:      JAM. JAM3. LN = MID$(IN$, 113, 3)
249:      JAM. ILLGLF = MID$(IN$, 116, 1)
250: END SUB
251:
252: '
253: '*****
254: '* センサス区間の指定 *
255: '* CALL CenseSel( EXEC$, RTNO%, CNSNO% ) *
256: '* EXEC$ ( 1 ) 処理名 *
257: '* RTNO% ( 0 ) 路線番号 *
258: '* CNSNO% ( 0 ) センサス区間番号 *
259: '*****
260: '
261: SUB CenseSel (EXEC$, RTNO%, CNSNO%)
262: '
263:     CLS
264:     PRINT "* "; EXEC$; " *"
265:     DO WHILE (1)
266:         CNSNO% = 0
267:         RTNO% = 0
268:         LOCATE 4, 1: PRINT SPACE$(80);
269:         CALL IntInput(1, 3, "路線番号を入力して下さい:", 4, 0, IN1%, 1
N$)
270:         IF (IN$ <> "") THEN
271:             CALL IntInput(1, 4, "センサス区間番号を入力して下さい:", 4, 0, IN2%, IN$)
272:             IF (IN$ <> "") THEN
273:                 RTNO% = IN1%
274:                 CNSNO% = IN2%
275:                 EXIT DO
276:             END IF
277:             ELSE
278:                 EXIT DO
279:             END IF
280:         LOOP
281: END SUB
282:
283: '
284: SUB CloseJamf
285:     CLOSE #1
286: END SUB
287:
288: '
289: '*****
290: '* 曜日算出 *
291: '* 0:日曜 1:月曜...7:土曜 *

```

```

292: '*****
293: '
294: FUNCTION DateWeek% (DT AS JamFdate)
295:     YY% = VAL(DT.YEAR) + 1900
296:     IF YY% < 1950 THEN YY% = YY% + 100
297:     MM% = VAL(DT.MON)
298:     DD% = VAL(DT.DAY)
299:     WW% = YY% - 1
300:     WW% = WW% + (INT(WW% / 4) - INT(WW% / 100) + INT(WW% / 400))
301:     ALLDAY% = 0
302:     IF (MM% = 2) THEN ALLDAY% = 31
303:     IF (MM% = 3) THEN ALLDAY% = 58
304:     IF (MM% = 4) THEN ALLDAY% = 90
305:     IF (MM% = 5) THEN ALLDAY% = 120
306:     IF (MM% = 6) THEN ALLDAY% = 151
307:     IF (MM% = 7) THEN ALLDAY% = 181
308:     IF (MM% = 8) THEN ALLDAY% = 212
309:     IF (MM% = 9) THEN ALLDAY% = 243
310:     IF (MM% = 10) THEN ALLDAY% = 273
311:     IF (MM% = 11) THEN ALLDAY% = 304
312:     IF (MM% = 12) THEN ALLDAY% = 334
313:     WW% = WW% + ALLDAY% + DD%
314:     IF (MM% > 2 AND LeapYear%(YY%) = 1) THEN WW% = WW% + 1
315:     TYP% = WW% MOD 7
316:
317:     IF (Holiday%(MM%, DD%) = 1) THEN
318:         TYP% = TYP% + 8
319:     ELSE
320:         IF (WW% = 1) THEN
321:             DD% = DD% - 1
322:             IF (DD% = 0) THEN
323:                 MM% = MM% - 1
324:                 IF (MM% = 0) THEN MM% = 12
325:                 DD% = 31
326:                 IF (MM% = 2) THEN
327:                     DD% = 28
328:                     IF (LeapYear%(YY%) = 1) THEN DD% = 29
329:                     ELSEIF (MM% = 4 OR MM% = 6 OR MM% = 9 OR MM% =
11) THEN
330:                         DD% = 30
331:                     END IF
332:                 END IF
333:                 IF (Holiday%(MM%, DD%) = 1) THEN TYP% = TYP% + 8
334:             END IF
335:         END IF
336:
337:         DateWeek% = TYP%
338:
339:     END FUNCTION
340:

```

```

341: '
342: '*****
343: '*   C R T画面座標軸の表示 (NO.1)   *
344: '*****
345: '
346: SUB DISP.AXIS1 (TILE$())
347: DEFINT X-Y
348: CLS
349: COLOR 3: X = 40: XX = 368: XD = 328: XB = X
350:
351: '---[BOX]---
352: FOR J = 1 TO 2
353:   LINE (XB, 320)-(XB + 260, 128), 3, B
354:   FOR I = 1 TO 5: LINE (XB, 128 + I * 32)-(XB + 2, 128 + I * 32), 3: NEXT
355:   FOR I = 1 TO 12: LINE (XB + I * 20, 318)-(XB + I * 20, 320), 3: NEXT
356:   XB = XB + XD
357: NEXT
358:
359: '---[TITLE]---
360: LOCATE 5, 6: PRINT "路線": LOCATE 5, 47: PRINT "路線"
361: LOCATE 5, 17: PRINT "センサ区間": LOCATE 5, 58: PRINT "センサ区間"
362: LOCATE 7, 6: PRINT "旅行速度時間変動"
363: LOCATE 7, 47: PRINT "渋滞長時間変動"
364: LOCATE 6, 30: PRINT "上り": LOCATE 6, 71: PRINT "上り"
365: LOCATE 7, 30: PRINT "下り": LOCATE 7, 71: PRINT "下り"
366: LINE (264, 88)-(288, 88), 6, , &HFFFF: LINE (264, 104)-(288, 104), 6, , &HCCCC
367: LINE (592, 86)-(616, 90), 6, B: PAINT (608, 88), TILE$(1), 6
368: LINE (592, 102)-(616, 106), 6, B: PAINT (608, 104), TILE$(2), 6
369:
370: '---[SCALE]---
371: ICOLOR% = 3
372:
373: FOR K = 1 TO 2: XB = 40: IF K = 2 THEN XB = 368
374:   FOR I = 1 TO 12: N = 6 + I: NN$ = STR$(N): NN = LEN(NN$) - 1
375:     X = XB + I * 20: YY = 322
376:     FOR J = 1 TO NN: XX = X - NN * 4 + (J - 1) * 8
377:       N$ = MID$(NN$, J + 1, 1): CALL PUT.N(XX, YY, N$, 1
COLOR%)
378:       NEXT J
379:     NEXT I
380:     K$ = "時   刻": FOR KK = 0 TO 4: KJ$ = MID$(K$, KK * 2 + 1, 2)
381:     XX = 80 + XB + KK * 20: YY = 336: CALL PUT.KJ(XX, YY, KJ$, ICOL
OR%)
382:     NEXT KK
383: NEXT K
384:
385: FOR I = 0 TO 6: N = I * 10: NN$ = STR$(N): NN = LEN(NN$) - 1
386:   X = 40 - NN * 8: YY = 318 - I * 32
387:   FOR J = 1 TO NN: XX = X + (J - 1) * 8
388:     N$ = MID$(NN$, J + 1, 1): CALL PUT.N(XX, YY, N$, ICOLOR%)

```

```

389:      NEXT J
390: NEXT I
391: K$ = "旅行速度"
392: FOR KK = 0 TO 3: KJ$ = MID$(K$, KK * 2 + 1, 2)
393:      XX = 5: YY = 160 + KK * 16: CALL PUT.KJ(XX, YY, KJ$, ICOLOR%)
394: NEXT KK
395: C$ = "km": FOR I = 1 TO 2: XX = 1 * 8: YY = 256: CH$ = MID$(C$, I, 1): CALL PU
      T.CH(XX, YY, CH$, ICOLOR%): NEXT
396: C$ = "/": XX = 10: YY = 272: CH$ = C$: CALL PUT.CH(XX, YY, CH$, ICOLOR%)
397: C$ = "h": XX = 10: YY = 288: CH$ = C$: CALL PUT.CH(XX, YY, CH$, ICOLOR%)
398:
399: FOR I = 0 TO 6: N = I: NN$ = STR$(N): NN = LEN(NN$) - 1
400:      X = 368 - NN * 8: YY = 318 - I * 32
401:      FOR J = 1 TO NN: XX = X + (J - 1) * 8
402:          N$ = MID$(NN$, J + 1, 1): CALL PUT.N(XX, YY, N$, ICOLOR%)
403:      NEXT J
404: NEXT I
405: K$ = "渋滞長"
406: FOR KK = 0 TO 2: KJ$ = MID$(K$, KK * 2 + 1, 2)
407:      XX = 336: YY = 160 + KK * 16: CALL PUT.KJ(XX, YY, KJ$, ICOLOR%)
408: NEXT KK
409: C$ = "km": FOR I = 1 TO 2: XX = 328 + I * 8: YY = 256: CH$ = MID$(C$, I, 1): C
      ALL PUT.CH(XX, YY, CH$, ICOLOR%): NEXT
410:
411: END SUB
412:
413: '
414: '*****
415: '*   C R T画面座標軸の表示   (NO.2)   *
416: '*****
417: '
418: SUB DISP.AXIS2 (TILE$())
419: DEFINT X-Y
420: CLS
421: COLOR 3: X = 40: XX = 368: XD = 328: XB = X
422:
423: '---[BOX]---
424: FOR J = 1 TO 2
425:     LINE (XB, 320)-(XB + 271, 128), 3, B
426:     FOR I = 1 TO 5: LINE (XB, 128 + I * 32)-(XB + 2, 128 + I * 32), 3: NEXT
427:     FOR I = 1 TO 13: LINE (XB + I * 20 - 5, 318)-(XB + I * 20 - 5, 320), 3: NEX
        T
428:     XB = XB + XD
429: NEXT
430:
431: '---[TITLE]---
432: LOCATE 5, 6: PRINT "路線": LOCATE 5, 47: PRINT "路線"
433: LOCATE 5, 17: PRINT "センサ区間": LOCATE 5, 58: PRINT "センサ区間"
434: LOCATE 7, 6: PRINT "旅行速度日変動"
435: LOCATE 7, 47: PRINT "渋滞長日変動"

```

```

436: LOCATE 6, 30: PRINT "上り": LOCATE 6, 71: PRINT "上り"
437: LOCATE 7, 30: PRINT "下り": LOCATE 7, 71: PRINT "下り"
438: LINE (264, 88)-(288, 88), 6, , &HFFFF: LINE (264, 104)-(288, 104), 6, , &HCCCC
439: LINE (592, 86)-(616, 90), 6, B: PAINT (608, 88), TILE$(1), 6
440: LINE (592, 102)-(616, 106), 6, B: PAINT (608, 104), TILE$(2), 6
441:
442: '---[SCALE]---
443: ICOLOR% = 3
444:
445: FOR K = 1 TO 2: XB = 48: IF K = 2 THEN XB = 376
446:     K$ = "日月火水木金土日平日平日平"
447:     FOR KK = 0 TO 12: KJ$ = MID$(K$, KK * 2 + 1, 2)
448:         XX = XB + KK * 20: YY = 322: CALL PUT.KJ(XX, YY, KJ$, ICOLOR%)
449:     NEXT KK
450:     K$ = " 3年秋    4冬4春4夏": FOR KK = 0 TO 11: KJ$ = MID$(K$, KK *
451:         2 + 1, 2)
452:         XX = XB + 20 + KK * 20: YY = 338: CALL PUT.KJ(XX, YY, KJ$, ICOL
453:         OR%)
454:     NEXT KK
455: NEXT K
456:
457: FOR I = 0 TO 6: N = I * 10: NN$ = STR$(N): NN = LEN(NN$) - 1
458:     X = 40 - NN * 8: YY = 318 - I * 32
459:     FOR J = 1 TO NN: XX = X + (J - 1) * 8
460:         N$ = MID$(NN$, J + 1, 1): CALL PUT.N(XX, YY, N$, ICOLOR%)
461:     NEXT J
462: NEXT I
463: K$ = "旅行速度"
464: FOR KK = 0 TO 3: KJ$ = MID$(K$, KK * 2 + 1, 2)
465:     XX = 5: YY = 160 + KK * 16: CALL PUT.KJ(XX, YY, KJ$, ICOLOR%)
466: NEXT KK
467: C$ = "km": FOR I = 1 TO 2: XX = I * 8: YY = 256: CH$ = MID$(C$, I, 1): CALL PU
468:     T.CH(XX, YY, CH$, ICOLOR%): NEXT
469: C$ = "/": XX = 10: YY = 272: CH$ = C$: CALL PUT.CH(XX, YY, CH$, ICOLOR%)
470: C$ = "h": XX = 10: YY = 288: CH$ = C$: CALL PUT.CH(XX, YY, CH$, ICOLOR%)
471:
472: FOR I = 0 TO 6: N = I: NN$ = STR$(N): NN = LEN(NN$) - 1
473:     X = 368 - NN * 8: YY = 318 - I * 32
474:     FOR J = 1 TO NN: XX = X + (J - 1) * 8
475:         N$ = MID$(NN$, J + 1, 1): CALL PUT.N(XX, YY, N$, ICOLOR%)
476:     NEXT J
477: NEXT I
478: K$ = "波滞長"
479: FOR KK = 0 TO 2: KJ$ = MID$(K$, KK * 2 + 1, 2)
480:     XX = 336: YY = 160 + KK * 16: CALL PUT.KJ(XX, YY, KJ$, ICOLOR%)
481: NEXT KK
482: C$ = "km": FOR I = 1 TO 2: XX = 328 + I * 8: YY = 256: CH$ = MID$(C$, I, 1): C
483:     ALL PUT.CH(XX, YY, CH$, ICOLOR%): NEXT
484:
485: END SUB

```

```

482:
483: '
484: '*****
485: '*   C R T画面座標軸の表示  (NO.3)   *
486: '*****
487: '
488: SUB DISP.AXIS3
489: DEFINT X-Y
490: CLS
491: COLOR 3: X = 48: XX = 368: XD = 328: XB = X
492:
493: '--[BOX]--
494: LINE (48, 320)-(288, 130), 3, B
495: LINE (368, 320)-(624, 128), 3, B
496: XB = X
497: FOR I = 1 TO 9: LINE (XB, 130 + I * 19)-(XB + 2, 130 + I * 19), 3: NEXT
498: FOR I = 1 TO 5: LINE (X + I * 40, 318)-(X + I * 40, 320), 3: NEXT
499:
500: XB = XX
501: FOR I = 1 TO 5: LINE (XB, 128 + I * 32)-(XB + 2, 128 + I * 32), 3: NEXT
502: FOR I = 1 TO 3: LINE (XX + I * 64, 318)-(XX + I * 64, 320), 3: NEXT
503:
504: '--[TITLE]--
505: LOCATE 5, 6: PRINT "路線": LOCATE 5, 47: PRINT "路線"
506: LOCATE 5, 17: PRINT "トンネル区間": LOCATE 5, 58: PRINT "トンネル区間"
507: LOCATE 7, 6: PRINT "旅行速度累加曲線"
508: LOCATE 7, 47: PRINT "Q-V図"
509: LOCATE 6, 30: PRINT "上り": LOCATE 6, 71: PRINT "上り"
510: LOCATE 7, 30: PRINT "下り": LOCATE 7, 71: PRINT "下り"
511: LINE (264, 88)-(288, 88), 6, , &HFFFF: LINE (264, 104)-(288, 104), 6, , &HCCCC
512: CIRCLE (600, 88), 2, 6: CIRCLE (600, 88), 1, 6: PSET (600, 88), 6
513: CIRCLE (600, 104), 2, 6
514:
515: '--[SCALE]--
516: ICOLOR% = 3
517:
518: XB = 40
519: FOR I = 0 TO 10: N = I * 10: NN$ = STR$(N): NN = LEN(NN$) - 1
520:     X = XB + 8 - NN * 8: YY = 318 - I * 19
521:     FOR J = 1 TO NN: XX = X + (J - 1) * 8
522:         N$ = MID$(NN$, J + 1, 1): CALL PUT.N(XX, YY, N$, ICOLOR%)
523:     NEXT J
524: NEXT I
525: K$ = "旅行速度"
526: FOR KK = 0 TO 3: KJ$ = MID$(K$, KK * 2 + 1, 2)
527:     XX = 100 + KK * 20: YY = 336: CALL PUT.KJ(XX, YY, KJ$, ICOLOR%)
528: NEXT KK
529: C$ = "km/h"
530: FOR KK = 0 TO 3: CH$ = MID$(C$, KK + 1, 1)
531:     XX = 210 + KK * 8: YY = 336: CALL PUT.CH(XX, YY, CH$, ICOLOR%)

```

```

532: NEXT KK
533: FOR I = 0 TO 6: N = I * 10: NN$ = STR$(N): NN = LEN(NN$) - 1
534:     X = 48 + I * 40: YY = 322
535:     FOR J = 1 TO NN: XX = X - NN * 4 + (J - 1) * 8
536:         N$ = MID$(NN$, J + 1, 1): CALL PUT.N(XX, YY, N$, ICOLOR%)
537:     NEXT J
538: NEXT I
539: K$ = "累加率 %": FOR KK = 0 TO 4: KJ$ = MID$(K$, KK * 2 + 1, 2)
540:     XX = XB - 28: YY = 180 + KK * 24: CALL PUT.KJ(XX, YY, KJ$, ICOLOR%)
541:     NEXT KK
542:
543: XB = 360
544: FOR I = 0 TO 6: N = I * 10: NN$ = STR$(N): NN = LEN(NN$) - 1
545:     X = XB + 8 - NN * 8: YY = 318 - I * 32
546:     FOR J = 1 TO NN: XX = X + (J - 1) * 8
547:         N$ = MID$(NN$, J + 1, 1): CALL PUT.N(XX, YY, N$, ICOLOR%)
548:     NEXT J
549: NEXT I
550: K$ = "旅行速度"
551: FOR KK = 0 TO 3: KJ$ = MID$(K$, KK * 2 + 1, 2)
552:     XX = XB - 28: YY = 160 + KK * 16: CALL PUT.KJ(XX, YY, KJ$, ICOLOR%)
553: NEXT KK
554: C$ = "km": FOR I = 1 TO 2: XX = XB - 35 + I * 8: YY = 256: CH$ = MID$(C$, I, 1)
555:     ): CALL PUT.CH(XX, YY, CH$, ICOLOR%): NEXT
556: C$ = "/": XX = XB - 25: YY = 272: CH$ = C$: CALL PUT.CH(XX, YY, CH$, ICOLOR%)
557: C$ = "h": XX = XB - 25: YY = 288: CH$ = C$: CALL PUT.CH(XX, YY, CH$, ICOLOR%)
558: FOR I = 0 TO 4: N = I * 500: NN$ = STR$(N): NN = LEN(NN$) - 1
559:     X = 368 + I * 64: YY = 322
560:     FOR J = 1 TO NN: XX = X - NN * 4 + (J - 1) * 8
561:         N$ = MID$(NN$, J + 1, 1): CALL PUT.N(XX, YY, N$, ICOLOR%)
562:     NEXT J
563: NEXT I
564: K$ = "1車線当り時間交通量 台/時": FOR KK = 0 TO 13: KJ$ = MID$(K$, KK * 2 + 1, 2)
565:     XX = 384 + KK * 16: YY = 336: CALL PUT.KJ(XX, YY, KJ$, ICOLOR%)
566:     NEXT KK
567:
568: END SUB
569:
570: '
571: '*****
572: '* CRT画面座標軸の表示 (NO.4) *
573: '*****
574: '
575: SUB DISP.AXIS4 (TILE$())
576: DEFINT X-Y
577: CLS
578: COLOR 3: XB = 48: YB = 336
579:

```

```

580: '---[BOX]--
581: LINE (XB, YB - 208)-(XB + 300, YB), 3, B
582: LINE (XB, YB - 104)-(XB + 300, YB - 104), 3
583: FOR I = 1 TO 25: LINE (XB, YB - I * 8)-(XB + 2, YB - I * 8), 3: NEXT
584:
585: '---[TITLE]--
586: LOCATE 5, 6: PRINT "路線"
587: LOCATE 5, 17: PRINT "ヒンサス区間"
588: LOCATE 7, 7: PRINT "渋滞発生状況図"
589: LOCATE 6, 33: PRINT "上り"
590: LOCATE 7, 33: PRINT "下り"
591: LINE (296, 86)-(320, 90), 6, B: PAINT (310, 88), TILE$(1), 6
592: LINE (296, 102)-(320, 106), 6, B: PAINT (310, 104), TILE$(2), 6
593:
594: '---[SCALE]--
595: ICOLOR% = 3
596:
597: FOR I = 1 TO 6: N = 6 + I * 2: NN$ = STR$(N): NN = LEN(NN$) - 1
598:     FOR J = 1 TO NN: XX = XB - NN * 8 + (J - 1) * 8
599:         NS$ = MID$(NN$, J + 1, 1)
600:         YY = YB - 108 + I * 16: CALL PUT.N(XX, YY, NS$, ICOLOR%)
601:         YY = YB - 108 - I * 16: CALL PUT.N(XX, YY, NS$, ICOLOR%)
602:     NEXT J
603: NEXT I
604: K$ = "起点": FOR KK = 0 TO 1: KJ$ = MID$(K$, KK * 2 + 1, 2)
605:     XX = XB - 32 + KK * 16: YY = YB - 112: CALL PUT.KJ(XX, YY, KJ$, ICOLOR%)
606: NEXT KK
607: K$ = "終点": FOR KK = 0 TO 1: KJ$ = MID$(K$, KK * 2 + 1, 2)
608:     XX = XB + 301 + KK * 16: YY = YB - 112: CALL PUT.KJ(XX, YY, KJ$, ICOLOR%)
609: NEXT KK
610:
611: K$ = "時刻"
612: FOR KK = 0 TO 5: KJ$ = MID$(K$, KK * 2 + 1, 2)
613:     XX = XB - 44: YY = YB - 152 + KK * 16: CALL PUT.KJ(XX, YY, KJ$, ICOLOR%)
614: NEXT KK
615:
616: END SUB
617:
618: DEFSNG X-Y
619: SUB DSORT (NDATA)
620: IF NDATA > 1 THEN
621:     FOR I% = 1 TO NDATA: FOR J% = I% TO NDATA
622:         I1A$ = LEFT$(INDT$(I%), 16): I2A$ = MID$(INDT$(I%), 22, 8)
623:         I3A$ = MID$(INDT$(I%), 20, 1)
624:         I1B$ = LEFT$(INDT$(J%), 16): I2B$ = MID$(INDT$(J%), 22, 8)
625:         I3B$ = MID$(INDT$(J%), 20, 1)
626:         IF I1A$ > I1B$ THEN

```

```

627:                                SWAP INDTS(1%), INDTS(J%)
628:                ELSE
629:                    IF 11A$ = 11B$ THEN
630:                        IF 12A$ > 12B$ THEN
631:                            SWAP INDTS(1%), INDTS(J%)
632:                        ELSE
633:                            IF 12A$ = 12B$ THEN
634:                                IF 13A$ > 13B$ THEN
635:                                    SWAP INDTS(1%), INDTS(J%)
636:                                    END IF
637:                                END IF
638:                            END IF
639:                        END IF
640:                    END IF
641:                NEXT J%, 1%
642:            END IF
643:
644:        END SUB
645:
646:    SUB DWGRAPH1 (TILE$, NDATA, CNSNO%, RTNO%, JAM AS JamF)
647:    DIM SPD!(2, 24), JMD!(2, 24), CNT%(2, 24)
648:
649:    CHKDAY$ = ""
650:    FOR KD% = 1 TO NDATA
651:        CALL BARA(INDTS(KD%), JAM)
652:        IN$ = MID$(INDTS(KD%), 22, 6)
653:        IF KD% = 1 THEN
654:            FOR I% = 1 TO 2: FOR J% = 1 TO 24
655:                SPD!(I%, J%) = 0: JMD!(I%, J%) = 0: CNT%(I%, J%) = 0
656:            NEXT J%, I%
657:            CHKDAY$ = IN$
658:            YY% = VAL(JAM.DATE.YEAR) + 1900
659:            IF YY% < 1950 THEN YY% = YY% + 100
660:            MM% = VAL(JAM.DATE.MON)
661:            DD% = VAL(JAM.DATE.DAY)
662:            WW% = (DateWeek%(JAM.DATE) MOD 7) + 1
663:        END IF
664:        IF IN$ <> CHKDAY$ THEN
665:            FOR I% = 1 TO 2: FOR J% = 1 TO 24
666:                N% = CNT%(I%, J%)
667:                IF N% <> 0 THEN
668:                    SPD!(I%, J%) = SPD!(I%, J%) / N%
669:                    JMD!(I%, J%) = JMD!(I%, J%) / N%
670:                END IF
671:            NEXT J%, I%
672:            CALL GRP1(TILE$, CNSNO%, RTNO%, YY%, MM%, DD%, WW%, SPD!(), J
MD!(), CNT%())
673:            IF COPY.SW% = 1 THEN
674:                CALL HCOPY(7, 0)
675:            ELSE

```

```

676:             WHILE INKEY$ = "": WEND
677:         END IF
678:         FOR I% = 1 TO 2: FOR J% = 1 TO 24
679:             SPD!(I%, J%) = 0: JMD!(I%, J%) = 0: CNT%(I%, J%) = 0
680:         NEXT J%, I%
681:         CHKDAY$ = IN$
682:         YY% = VAL(JAM.DATE.YEAR) + 1900
683:         IF YY% < 1950 THEN YY% = YY% + 100
684:         MM% = VAL(JAM.DATE.MON)
685:         DD% = VAL(JAM.DATE.DAY)
686:         WW% = (DateWeek%(JAM.DATE) MOD 7) + 1
687:     END IF
688:
689:     TIMIDX% = VAL(JAM.DATE.HH)
690:     IF (TIMIDX% >= 24) THEN TIMIDX% = 0
691:     TIMIDX% = TIMIDX% + 1
692:     UPDOWN% = VAL(JAM.DIRECT)
693:     IF (UPDOWN% <> 1 AND UPDOWN% <> 2) THEN UPDOWN% = 1
694:     SPEED! = VAL(JAM.SPEED) / 10!
695:     LTOT! = VAL(JAM.JAMLEN1) * 10! + VAL(JAM.JAMLEN2) * 10!
696:     SPD!(UPDOWN%, TIMIDX%) = SPD!(UPDOWN%, TIMIDX%) + SPEED!
697:     JMD!(UPDOWN%, TIMIDX%) = JMD!(UPDOWN%, TIMIDX%) + LTOT!
698:     CNT%(UPDOWN%, TIMIDX%) = CNT%(UPDOWN%, TIMIDX%) + 1
699: NEXT KD%
700:
701: FOR I% = 1 TO 2: FOR J% = 1 TO 24
702:     N% = CNT%(I, J)
703:     IF N% <> 0 THEN
704:         SPD!(I, J) = SPD!(I, J) / N%
705:         JMD!(I, J) = JMD!(I, J) / N%
706:     END IF
707: NEXT J%, I%
708: CALL GRP1(TILE$(), CNSNO%, RTNO%, YY%, MM%, DD%, WW%, SPD!(), JMD!(), CNT%())
709: IF COPY.SW% = 1 THEN
710:     CALL HCOPY(7, 0)
711: ELSE
712:     WHILE INKEY$ = "": WEND
713: END IF
714:
715: END SUB
716:
717: SUB DWGRAPH2 (TILE$(), NDATA, CNSNO%, RTNO%, SMAXMIN!(), JMAXMIN!())
718: DIM LPTRN%(2)
719: '---- 線・タイルパターン設定 ----
720: LPTRN%(1) = &HFFFF
721: LPTRN%(2) = &HCCCC
722:
723: CALL DISP.AXIS2(TILE$())
724: LOCATE 5, 11: PRINT USING "####": RTNO%
725: LOCATE 5, 26: PRINT USING "####": CNSNO%

```

```

726: LOCATE 5, 52: PRINT USING "####"; RTNO%
727: LOCATE 5, 67: PRINT USING "####"; CNSNO%
728:
729: '---日別旅行速度---
730: YB% = 320: XB% = 35: ICOLOR% = 2
731: FOR I% = 1 TO 2
732:     IGRPH% = 0: IDX% = -3: IF I% = 2 THEN IDX% = 3
733:     FOR J% = 1 TO 13
734:         IX% = XB% + J% * 20 + IDX%
735:         SMX! = SMAXMIN!(J%, I%, 1): SMI! = SMAXMIN!(J%, I%, 2): SAV! = SMAXMIN!
(J%, I%, 3)
736:         JY1% = YB% - SMX! * 3.2: JY2% = YB% - SMI! * 3.2: JY3% = YB% - SAV! * 3
.2
737:         IF JY1% < 128 THEN JY1% = 128
738:         IF (SAV! <> 0!) THEN
739:             IY2% = JY3%
740:             CIRCLE (IX%, IY2%), 2, ICOLOR%
741:             IF (SMX! <> SMI!) THEN
742:                 LINE (IX% + 2, JY1%)-(IX% - 2, JY1%), ICOLOR%
743:                 LINE (IX% + 2, JY2%)-(IX% - 2, JY2%), ICOLOR%
744:                 LINE (IX%, JY1%)-(IX%, JY2%), ICOLOR%, , LPTRN%(I%)
745:             END IF
746:             IF (IGRPH% <> 0) THEN
747:                 LINE (IX1%, IY1%)-(IX%, IY2%), ICOLOR%, , LPTRN%(I%)
748:             END IF
749:             IGRPH% = 1
750:             IF (SAV! > 60!) THEN IGRPH% = 0
751:             IY1% = IY2%: IX1% = IX%
752:         END IF
753:     NEXT J%
754: NEXT I%
755:
756: '---日別渋滞長---
757: XB% = 365
758: FOR I% = 1 TO 2
759:     IDX% = -6: IF I% = 2 THEN IDX% = 6
760:     FOR J% = 1 TO 13
761:         IX% = XB% + J% * 20
762:         JMX! = JMAXMIN!(J%, I%, 1): JMI! = JMAXMIN!(J%, I%, 2): JAV! = JMAXMIN!
(J%, I%, 3)
763:         JY1% = YB% - JMX! * .032: JY2% = YB% - JMI! * .032: JY3% = YB% - JAV! *
.032
764:         IF JY1% < 128 THEN JY1% = 128
765:         IF (JAV! <> 0) THEN
766:             IF (JMX! <> JMI!) THEN
767:                 LINE (IX% + IDX%, JY1%)-(IX%, JY2%), ICOLOR%, B
768:                 IF JY2% - JY1% > 2 THEN
769:                     PAINT (IX% + IDX% / 2, (JY1% + JY2%) / 2), TILE
$(I%), ICOLOR%
770:             END IF

```

```

771:                END IF
772:                LINE (IX% + IDX, JY3% + 1)-(IX%, JY3%), ICOLOR%, BF
773:            END IF
774:        NEXT J%
775:    NEXT I%
776:
777:    IF COPY.SW% = 1 THEN
778:        CALL HCOPY(7, 0)
779:    ELSE
780:        WHILE INKEY$ = "": WEND
781:    END IF
782:
783:    END SUB
784:
785:    SUB DWGRAPH3 (NDATA, CNSNO%, RTNO%, SPDSUM!())
786:    DIM LPTRN%(2)
787:    '--- 線・タイルパターン設定 ---
788:    LPTRN%(1) = &HFFFF
789:    LPTRN%(2) = &HCCCC
790:    CALL DISP.AXIS3
791:    LOCATE 5, 11: PRINT USING "####": RTNO%
792:    LOCATE 5, 26: PRINT USING "####": CNSNO%
793:    LOCATE 5, 52: PRINT USING "####": RTNO%
794:    LOCATE 5, 67: PRINT USING "####": CNSNO%
795:
796:    '---旅行速度累加曲線---
797:    YB% = 320: XB% = 48
798:    FOR ID% = 1 TO 2
799:        PSET (XB%, YB%), 2
800:        FOR I% = 1 TO 60
801:            IX% = XB% + I% * 4!
802:            SPD! = SPDSUM!(ID%, I%)
803:            IY% = YB% - SPD! * 1.9
804:            LINE -(IX%, IY%), 2, , LPTRN%(ID%)
805:        NEXT I%
806:    NEXT ID%
807:
808:    CALL GRP3(NDATA)
809:
810:    IF COPY.SW% = 1 THEN
811:        CALL HCOPY(7, 0)
812:    ELSE
813:        WHILE INKEY$ = "": WEND
814:    END IF
815:
816:    END SUB
817:
818:    SUB DWGRAPH4 (TILE$(), NDATA, CNSNO%, RTNO%, JAM AS JamF)
819:
820:    DIM JPOS!(2, 24, 3), JLEN!(2, 24, 3), CNT%(2, 24, 3)

```

```

821:
822: ALLDIS! = VAL(MID$(INDT$(1), 17, 3)) / 10
823: UNI! = 300 / ALLDIS!
824:
825: CHKDAY$ = ""
826: FOR KD% = 1 TO NDATA
827:     CALL BARA(INDT$(KD%), JAM)
828:     IN$ = MID$(INDT$(KD%), 22, 6)
829:     IF KD% = 1 THEN
830:         FOR I% = 1 TO 2: FOR J% = 1 TO 24: FOR K% = 1 TO 3
831:             JPOS!(I%, J%, K%) = 0: JLEN!(I%, J%, K%) = 0: CNT%(I%,
J%, K%) = 0
832:         NEXT K%, J%, I%
833:         CHKDAY$ = IN$
834:         YY% = VAL(JAM. DATE. YEAR) + 1900
835:         IF YY% < 1950 THEN YY% = YY% + 100
836:         MM% = VAL(JAM. DATE. MON)
837:         DD% = VAL(JAM. DATE. DAY)
838:         WW% = (DateWeek%(JAM. DATE) MOD 7) + 1
839:     END IF
840:     IF IN$ <> CHKDAY$ THEN
841:         FOR I% = 1 TO 2: FOR J% = 1 TO 24: FOR K% = 1 TO 3
842:             N% = CNT%(I%, J%, K%)
843:             IF N% <> 0 THEN
844:                 JPOS!(I%, J%, K%) = JPOS!(I%, J%, K%) / N%
845:                 JLEN!(I%, J%, K%) = JLEN!(I%, J%, K%) / N%
846:                 IF I% = 1 THEN JPOS!(I%, J%, K%) = ALLDIS! - JP
OS!(I%, J%, K%) - JLEN!(I%, J%, K%)
847:             END IF
848:         NEXT K%, J%, I%
849:         CALL GRP4(TILE$( ), UNI!, CNSNO%, RTNO%, YY%, MM%, DD%, WW%, JPO
S!( ), JLEN!( ), CNT%( ))
850:         IF COPY.SW% = 1 THEN
851:             CALL HCOPY(7, 0)
852:         ELSE
853:             WHILE INKEY$ = "": WEND
854:         END IF
855:         FOR I% = 1 TO 2: FOR J% = 1 TO 24: FOR K% = 1 TO 3
856:             JPOS!(I%, J%, K%) = 0: JLEN!(I%, J%, K%) = 0: CNT%(I%,
J%, K%) = 0
857:         NEXT K%, J%, I%
858:         CHKDAY$ = IN$
859:         YY% = VAL(JAM. DATE. YEAR) + 1900
860:         IF YY% < 1950 THEN YY% = YY% + 100
861:         MM% = VAL(JAM. DATE. MON)
862:         DD% = VAL(JAM. DATE. DAY)
863:         WW% = (DateWeek%(JAM. DATE) MOD 7) + 1
864:     END IF
865:
866: TIMIDX% = VAL(JAM. DATE. HH)

```

```

867: IF (TIMIDX% >= 24) THEN TIMIDX% = 0
868: TIMIDX% = TIMIDX% + 1
869: UPDWN% = VAL(JAM.DIRECT)
870: IF (UPDWN% <> 1 AND UPDWN% <> 2) THEN UPDWN% = 1
871: JL! = VAL(JAM.JAM1.LN) / 10
872: JPOS!(UPDWN%, TIMIDX%, 1) = VAL(JAM.JAM1.PS) / 10
873: JLEN!(UPDWN%, TIMIDX%, 1) = JL!
874: IF JL! <> 0 THEN CNT%(UPDWN%, TIMIDX%, 1) = CNT%(UPDWN%, TIMIDX%, 1) + 1
875: JL! = VAL(JAM.JAM2.LN) / 10
876: JPOS!(UPDWN%, TIMIDX%, 2) = VAL(JAM.JAM2.PS) / 10
877: JLEN!(UPDWN%, TIMIDX%, 2) = JL!
878: IF JL! <> 0 THEN CNT%(UPDWN%, TIMIDX%, 2) = CNT%(UPDWN%, TIMIDX%, 2) + 1
879: JL! = VAL(JAM.JAM3.LN) / 10
880: JPOS!(UPDWN%, TIMIDX%, 3) = VAL(JAM.JAM3.PS) / 10
881: JLEN!(UPDWN%, TIMIDX%, 3) = JL!
882: IF JL! <> 0 THEN CNT%(UPDWN%, TIMIDX%, 3) = CNT%(UPDWN%, TIMIDX%, 3) + 1
883: NEXT KD%
884:
885: FOR I% = 1 TO 2: FOR J% = 1 TO 24: FOR K% = 1 TO 3
886:     N% = CNT%(I%, J%, K%)
887:     IF N% <> 0 THEN
888:         JPOS!(I%, J%, K%) = JPOS!(I%, J%, K%) / N%
889:         JLEN!(I%, J%, K%) = JLEN!(I%, J%, K%) / N%
890:         IF I% = 1 THEN JPOS!(I%, J%, K%) = ALLDIS! - JPOS!(I%, J%, K%)
            - JLEN!(I%, J%, K%)
891:     END IF
892: NEXT K%, J%, I%
893: CALL GRP4(TILE$, UNI!, CNSNO%, RTNO%, YY%, MM%, DD%, WN%, JPOS!(), JLEN!(),
    CNT%())
894: IF COPY.SW% = 1 THEN
895:     CALL HCOPY(7, 0)
896: ELSE
897:     WHILE INKEY$ = "": WEND
898: END IF
899:
900: END SUB
901:
902: SUB GRP1 (TILE$(), CNSNO%, RTNO%, YY%, MM%, DD%, WN%, SPD!(), JMD!(), CNT%())
903: DIM LPTRN%(2), WEEK$(8)
904: '--- 線・タイルパターン設定 ---
905: LPTRN%(1) = &HFFFF
906: LPTRN%(2) = &HCCCC
907: WEEK$(1) = "日": WEEK$(2) = "月": WEEK$(3) = "火": WEEK$(4) = "水"
908: WEEK$(5) = "木": WEEK$(6) = "金": WEEK$(7) = "土"
909:
910: CALL DISP.AXIS1(TILE$())
911: LOCATE 5, 11: PRINT USING "####": RTNO%
912: LOCATE 5, 26: PRINT USING "####": CNSNO%
913: LOCATE 6, 6: PRINT USING "####年##月##日 (&&) "; YY%; MM%; DD%; WEEK$(WN%);
914: LOCATE 5, 52: PRINT USING "####": RTNO%

```

```

915: LOCATE 5, 67: PRINT USING "####"; CNSNO%
916: LOCATE 6, 47: PRINT USING "####年##月##日 (&&) "; YY%; MM%; DD%; WEEK$(NN%);
917:
918: '---時刻別旅行速度---
919: YB% = 320: XB% = 40: ICOLOR% = 2
920: FOR I% = 1 TO 2
921:     IGRPH% = 0
922:     FOR JJ% = 7 TO 18: J% = JJ% + 1
923:     IX% = XB% + (J% - 7) * 20
924:     IF (CNT%(I%, J%) > 0) THEN
925:         SPD! = SPD!(I%, J%)
926:         IF (SPD! > 60!) THEN SPD! = 60!
927:         IY2% = YB% - SPD! * 3.2
928:         CIRCLE (IX%, IY2%), 1, 5
929:         IF (IGRPH% <> 0) THEN
930:             LINE (IX1%, IY1%)-(IX%, IY2%), ICOLOR%, , LPTRN%(I%)
931:         END IF
932:         IGRPH% = 1
933:         IF (SPD!(I%, J%) > 60!) THEN IGRPH% = 0
934:         IY1% = IY2%: IX1% = IX%
935:     END IF
936: NEXT JJ%
937: NEXT I%
938:
939: '---時間帶別波滯長---
940: XB% = 368
941: FOR I% = 1 TO 2
942:     IDX% = -6: IF I% = 2 THEN IDX% = 6
943:     FOR JJ% = 7 TO 18: J% = JJ% + 1
944:     IX% = XB% + (J% - 7) * 20
945:     IF (CNT%(I%, J%) > 0) THEN
946:         JLN! = JMD!(I%, J%) / 1000
947:         IF (JLN! > 6!) THEN JLN! = 6!
948:         IY2% = YB% - JLN! * 32
949:         LINE (IX% + IDX%, IY2%)-(IX%, YB%), ICOLOR%, B
950:         IF YB% - IY2% > 2 THEN
951:             PAINT (IX% + IDX% / 2, (YB% + IY2%) / 2), TILE$(I%), IC
952:             OLOR%
953:         END IF
954:     END IF
955: NEXT JJ%
956: NEXT I%
957: END SUB
958:
959: SUB GRP3 (NDATA)
960:
961: '---Q V曲線---
962: XB% = 368: YB% = 320
963: FOR KD% = 1 TO NDATA

```

```

964:      VOL! = VAL(MID$(INDT$(KD%), 45, 4))
965:      SPD! = VAL(MID$(INDT$(KD%), 62, 4)) / 10
966:      WW! = VAL(MID$(INDT$(KD%), 21, 1))
967:      UP% = VAL(MID$(INDT$(KD%), 20, 1))
968:      IF WW! <> 0 THEN VOL! = VOL! / WW!
969:      IF UP% <> 1 AND UP% <> 2 THEN UP% = 1
970:
971:      IF SPD! < 60 THEN
972:      IF VOL! > .1 THEN
973:          IX% = XB% + VOL! / 500 * 64
974:          IY% = YB% - SPD! * 3.2
975:          CIRCLE (IX%, IY%), 2, 2
976:          IF UP% = 1 THEN
977:              CIRCLE (IX%, IY%), 1, 2: PSET (IX%, IY%), 2
978:          END IF
979:      END IF
980:  END IF
981: NEXT KD%
982:
983: END SUB
984:
985: SUB GRP4 (TILE$, UNI!, CNSNO%, RTNO%, YY%, MM%, DD%, WW%, JPOS!(), JLEN!(),
CNT%())
986: DIM WEEK$(8)
987: WEEK$(1) = "日": WEEK$(2) = "月": WEEK$(3) = "火": WEEK$(4) = "水"
988: WEEK$(5) = "木": WEEK$(6) = "金": WEEK$(7) = "土"
989:
990: CALL DISP.AXIS4(TILE$())
991: LOCATE 5, 11: PRINT USING "####": RTNO%
992: LOCATE 5, 26: PRINT USING "####": CNSNO%
993: LOCATE 6, 6: PRINT USING "####年##月##日 (&&) "; YY%; MM%; DD%; WEEK$(WW%);
994:
995: '---渋滞状況図---
996: YB% = 336: XB% = 44: DY% = 106: IW% = 5: ICOLOR% = 2
997: FOR I% = 1 TO 2
998:     IDY% = 8: IF I% = 2 THEN IDY% = -8
999:     FOR JJ% = 7 TO 18: J% = JJ% + 1
1000:     IY% = YB% - DY% + (J% - 7) * IDY%
1001:     FOR K% = 1 TO 3
1002:         IF (CNT%(I%, J%, K%) > 0) THEN
1003:             JP! = JPOS!(I%, J%, K%)
1004:             JL! = JLEN!(I%, J%, K%)
1005:             IX1% = XB% + JP! * UNI!: IX2% = IX1% + JL! * UNI!
1006:             LINE (IX1%, IY% + IW%)-(IX2%, IY%), ICOLOR%, B
1007:             IF YB% - IY2% > 2 THEN
1008:                 PAINT ((IX1% + IX2%) / 2, IY% + IW% / 2), TILE$
1009:                 (I%), ICOLOR%
1010:             END IF
1011:         END IF
1012:     NEXT K%
1013: NEXT JJ%
1014: NEXT I%

```

```

1012:      NEXT JJ%
1013: NEXT I%
1014:
1015: IX% = XB% + UNI!: IY% = YB% - DY%
1016: WHILE IX% < 352
1017:      LINE (IX%, IY% + 4)-(IX%, IY% - 1), 2
1018:      IX% = IX% + UNI!
1019: WEND
1020:
1021: END SUB
1022:
1023: '*****
1024: '* 祝日判定 *
1025: '* 0:平日 1:祝日 *
1026: '*****
1027: '
1028: FUNCTION Holiday% (MM%, DD%)
1029: '
1030:      Holiday% = 0
1031:
1032:      RESTORE HolidayData
1033:      FOR I% = 1 TO MM% - 1
1034:          READ ICNT%
1035:          FOR J% = 1 TO ICNT%
1036:              READ IDMY%
1037:          NEXT J%
1038:      NEXT I%
1039:
1040:      READ ICNT%
1041:      FOR I% = 1 TO ICNT%
1042:          READ J%
1043:          IF (J% = DD%) THEN
1044:              Holiday% = 1
1045:              EXIT FOR
1046:          END IF
1047:      NEXT I%
1048:
1049: END FUNCTION
1050:
1051: '
1052: '*****
1053: '* 正の整数数値入力 *
1054: '*****
1055: '
1056: SUB IntInput (IX%, IY%, PRMP$, ILN%, IWAIT%, IN%, IN$)
1057: '
1058:      IN% = 0
1059:      DO WHILE (1)
1060:          CALL LineEdit (IX%, IY%, PRMP$, ILN%, 1, IWAIT%, IN$)
1061:

```

```

1062:          IF (IN$ = "") THEN EXIT DO
1063: '
1064:          L% = LEN(IN$)
1065:          IST% = 0
1066:          FOR I% = 1 TO L%
1067:              IS$ = MID$(IN$, I%, 1)
1068:              IF (IS$ < "0" OR IS$ > "9") THEN
1069:                  IST% = 1
1070:                  EXIT FOR
1071:              END IF
1072:          NEXT I%
1073: '
1074:          IF (IST% = 0) THEN
1075:              IN% = VAL(IN$)
1076:              EXIT DO
1077:          END IF
1078:          BEEP
1079:      LOOP
1080: END SUB
1081:
1082: '
1083: '*****
1084: '* 閏年判定 *
1085: '* 0:平年 1:閏年 *
1086: '*****
1087: '
1088: FUNCTION LeapYear% (YY%)
1089: '
1090:     LeapYear% = ((YY% MOD 4) = 0) AND (((YY% MOD 100) <> 0) OR ((YY% MOD 40
0) = 0))
1091: '
1092: END FUNCTION
1093:
1094: '
1095: '*****
1096: '* 1行編集入力 *
1097: '*****
1098: '
1099: SUB LineEdit (IX%, IY%, PRMP$, ILN%, INEW%, IWAIT%, IN$)
1100: '
1101:     IF (INEW% = 1) THEN IN$ = ""
1102: '
1103:     LOCATE IY%, IX%
1104:     PRINT PRMP$;
1105: '
1106:     IPRL% = LEN(PRMP$)
1107:     ILNWK% = ILN%
1108:     IF (ILN% = 1) THEN ILNWK% = 2
1109:     LOCATE IY%, IX% + IPRL%
1110:     PRINT SPACES(80 - IPRL%);

```

```

1111: '
1112:     DO WHILE (1)
1113:         LOCATE IY%, IX% + IPRL%
1114:         COLOR 3
1115:         IN$ = Edit$(IN$, ILNWK%, LK$, 0)
1116:         COLOR 3
1117:
1118:         IF (LK$ = CHR$(&HD)) THEN
1119:             IST% = 0
1120:             IF (ILN% = 1) THEN
1121:                 IF (LEN(IN$) > 1) THEN
1122:                     IST% = 1
1123:                     IN$ = ""
1124:                     BEEP
1125:                 END IF
1126:             END IF
1127:             IF (IST% = 0) THEN
1128:                 IF (IN$ <> "") THEN EXIT DO
1129:                 IF (IN$ = "" AND IWAIT% = 0) THEN EXIT DO
1130:             END IF
1131:         END IF
1132:     LOOP
1133: '
1134: END SUB
1135:
1136: '
1137: '*****
1138: '*   波滞情報ファイル関係サブルーチン   *
1139: '*****
1140: '
1141: SUB OpenJamf (UNIT$, JamF$)
1142:     OPEN UNIT$ + JamF$ FOR INPUT AS #1
1143: END SUB
1144:
1145: SUB PR.oneline (IN$)
1146:
1147:     S$ = " "
1148:     OUT$ = SPACE$(26)
1149:     OUT$ = OUT$ + MID$(IN$, 20, 1) + S$
1150:     OUT$ = OUT$ + MID$(IN$, 22, 8) + S$ + MID$(IN$, 30, 1) + S$ + MID$(IN$,
31, 1) + S$
1151:     OUT$ = OUT$ + MID$(IN$, 32, 1) + S$ + MID$(IN$, 33, 6) + S$ + MID$(IN$,
39, 6) + S$
1152:     OUT$ = OUT$ + MID$(IN$, 45, 4) + S$ + MID$(IN$, 49, 4) + S$ + MID$(IN$,
53, 6) + S$
1153:     OUT$ = OUT$ + MID$(IN$, 59, 3) + S$ + MID$(IN$, 62, 4) + S$ + MID$(IN$,
66, 4) + S$
1154:     OUT$ = OUT$ + MID$(IN$, 70, 4) + S$ + MID$(IN$, 74, 4) + S$ + MID$(IN$,
78, 4) + S$
1155:     OUT$ = OUT$ + MID$(IN$, 82, 4) + S$ + MID$(IN$, 86, 4) + S$ + MID$(IN$,

```

```

90, 4) + S$
1156: OUT$ = OUT$ + MID$(IN$, 94, 4) + S$ + MID$(IN$, 98, 3) + S$ + MID$(IN$,
101, 3)
1157: PRINT #2, OUT$
1158:
1159: END SUB
1160:
1161: '*****
1162: '* 渋滞指標の印字 *
1163: '*****
1164: SUB PR.Outline (NDATA, JAM AS JamF)
1165: DIM SDATA(2, 13), NNDD(2, 13)
1166:
1167: F1$ = "          調  調  天視路 通過時刻      交通量  旅
          停 旅 平  平加 停  総渋滞長 渋滞区間 渋旅 最大 "
1168: F2$ = "          査  査  面      (台/時)  行
          止 行 均加 均加 止  (10m)  通過時間 滞行 渋滞長"
1169: F3$ = "          方  日  候界状 起 終      時
          回 速  速  速  時      (秒)  区速  点  点
1170: F4$ = "          向      況  点  点      間
          数 度  度  度  間 ラ ラ ラ ラ 間度 位 長"
1171: F5$ = "
          シ シ シ シ 内 置 さ"
1172: F6$ = "          年月日時      時分秒 時分秒 全車 大型 時分
          秒 0.1 0.01 0.01 秒 ク ク ク ク 0.1 100 100"
1173: F7$ = "
          km/h m/s2 m/s3 1 2 1 2 km/h m m "
1174:
1175:
1176: 'NLINE% = 50
1177:
1178: IF COPY.SW% = 1 THEN
1179: FOR I% = 1 TO 2: FOR J% = 1 TO 13
1180: SDATA(I%, J%) = 0: NNDD(I%, J%) = 0
1181: NEXT J%, I%
1182:
1183: FOR I% = 1 TO NDATA
1184: IN$ = INDT$(I%)
1185:
1186: IF I% = 1 THEN
1187: IF NLINE% > 47 THEN
1188: F8$ = SPACES$(26) + "地域コード " + LEFT$(IN$, 5)
1189: F8$ = F8$ + " 調査単位区間番号 " + MID$(IN$, 6, 6)
1190: F8$ = F8$ + " 路線番号 " + MID$(IN$, 12, 4) + " 道路種別 " + MI
D$(IN$, 16, 1)
1191: F8$ = F8$ + " 区間長 " + MID$(IN$, 17, 3) + "00 m 車線数 " + MI
D$(IN$, 21, 1)
1192: PRINT #2, CHR$(12)
1193: PRINT #2, "": PRINT #2, "": PRINT #2, "": PRINT #2, ""
1194: PRINT #2, F8$: PRINT #2, ""

```

```

1195:          PRINT #2, F1$: PRINT #2, F2$: PRINT #2, F3$: PRINT #2, F4$
1196:          PRINT #2, F5$: PRINT #2, F6$: PRINT #2, F7$
1197:          PRINT #2, " "
1198:      '      NLINE% = 8
1199:      END IF
1200:
1201: IF I% = 1 THEN CHKDAY$ = MID$(IN$, 22, 6)
1202:
1203: IF MID$(IN$, 22, 6) = CHKDAY$ THEN
1204:     CALL PR.oneline(IN$)
1205:     '      NLINE% = NLINE% + 1
1206:     CALL Sumdata2(IN$, SDATA(), NNDD(), JAM)
1207: ELSE
1208:     CALL PR.Sumline(SDATA(), NNDD())
1209:     '      NLINE% = NLINE% + 3
1210:     CHKDAY$ = MID$(IN$, 22, 6)
1211:     F8$ = SPACE$(26) + "地域コード " + LEFT$(IN$, 5)
1212:     F8$ = F8$ + " 調査単位区間番号 " + MID$(IN$, 6, 6)
1213:     F8$ = F8$ + " 路線番号 " + MID$(IN$, 12, 4) + " 道路種別 " + MI
D$(IN$, 16, 1)
1214:     F8$ = F8$ + " 区間長 " + MID$(IN$, 17, 3) + "00 m 車線数 " + MI
D$(IN$, 21, 1)
1215:     PRINT #2, CHR$(12)
1216:     PRINT #2, "": PRINT #2, "": PRINT #2, "": PRINT #2, ""
1217:     PRINT #2, F8$: PRINT #2, ""
1218:     PRINT #2, F1$: PRINT #2, F2$: PRINT #2, F3$: PRINT #2, F4$
1219:     PRINT #2, F5$: PRINT #2, F6$: PRINT #2, F7$
1220:     PRINT #2, " "
1221:     CALL PR.oneline(IN$)
1222:     '      NLINE% = NLINE% + 1
1223:     CALL Sumdata2(IN$, SDATA(), NNDD(), JAM)
1224: END IF
1225: NEXT I%
1226: CALL PR.Sumline(SDATA(), NNDD())
1227: PRINT #2, CHR$(12)
1228: END IF
1229:
1230: END SUB
1231:
1232: SUB PR.Sumline (SDATA(), ND())
1233: DIM HED$(2)
1234: FOR I% = 1 TO 2: FOR J% = 1 TO 13
1235:     NN% = ND(I%, J%)
1236:     IF NN% <> 0 THEN
1237:         S% = SDATA(I%, J%) / NN%
1238:         SDATA(I%, J%) = S%
1239:     ELSE
1240:         SDATA(I%, J%) = 0
1241:     END IF
1242: NEXT J%

```

```

1243: NEXT I%
1244:
1245: HED$(1) = "小計 (日平均値)      上り" + SPACES$(5)
1246: HED$(2) = "                      下り" + SPACES$(5)
1247: FOR I% = 1 TO 2
1248: OUT$ = SPACES$(26) + HED$(I%) + RIGHT$("      " + STR$(SDATA(I%, 1)), 4)
1249: OUT$ = OUT$ + RIGHT$("      " + STR$(SDATA(I%, 2)), 5)
1250: HH% = INT(SDATA(I%, 3) / 3600)
1251: OUT$ = OUT$ + RIGHT$("      " + STR$(HH%), 3)
1252: MM% = INT(SDATA(I%, 3) - HH% * 3600) / 60
1253: OUT$ = OUT$ + RIGHT$("      " + STR$(MM%), 2)
1254: SS% = SDATA(I%, 3) MOD 60
1255: OUT$ = OUT$ + RIGHT$("      " + STR$(SS%), 2)
1256: OUT$ = OUT$ + RIGHT$("      " + STR$(SDATA(I%, 4)), 4)
1257: OUT$ = OUT$ + RIGHT$("      " + STR$(SDATA(I%, 5)), 5)
1258: OUT$ = OUT$ + RIGHT$("      " + STR$(SDATA(I%, 6)), 5)
1259: OUT$ = OUT$ + RIGHT$("      " + STR$(SDATA(I%, 7)), 5)
1260: OUT$ = OUT$ + RIGHT$("      " + STR$(SDATA(I%, 8)), 5)
1261: OUT$ = OUT$ + RIGHT$("      " + STR$(SDATA(I%, 9)), 5)
1262: OUT$ = OUT$ + RIGHT$("      " + STR$(SDATA(I%, 10)), 5)
1263: OUT$ = OUT$ + RIGHT$("      " + STR$(SDATA(I%, 11)), 5)
1264: OUT$ = OUT$ + RIGHT$("      " + STR$(SDATA(I%, 12)), 5)
1265: OUT$ = OUT$ + RIGHT$("      " + STR$(SDATA(I%, 13)), 5)
1266: PRINT #2, OUT$
1267: NEXT I%
1268: PRINT #2, " "
1269:
1270: FOR I% = 1 TO 2: FOR J% = 1 TO 13
1271:     SDATA(I%, J%) = 0: ND(I%, J%) = 0
1272: NEXT J%, I%
1273:
1274: END SUB
1275:
1276: '
1277: SUB ReadJamf (NDATA, RTNO%, CNSNO%, JAM AS JamF)
1278: DIM DTA AS DtaType
1279: FUNC% = 0: NDATA = 0
1280: DO WHILE (1)
1281:     FPATH$ = UNIT$ + "*.SUM"
1282:     JamF$ = GetFile$(FPATH$, ATR%, DTA, FUNC%)
1283:     FUNC% = 1
1284:     IF (JamF$ = "") THEN EXIT DO
1285:
1286:     CALL OpenJamf(UNIT$, JamF$)
1287:     DO WHILE (1)
1288:         IF (EOF(1)) THEN EXIT DO
1289:         LINE INPUT #1, IN$
1290:         CALL BARA(IN$, JAM)
1291:         IF (VAL(JAM.CODE.CENSNO) = CNSNO% AND VAL(JAM.RTNO) = RTNO%) TH

```

EN

```

1292:                                NDATA = NDATA + 1
1293:                                INDTS(NDATA) = INS
1294:                                END IF
1295:                                LOOP
1296:                                CALL CloseJamf
1297: LOOP
1298: END SUB
1299:
1300: '
1301: '*****
1302: '*   渋滞指標の集計   *
1303: '*****
1304: SUB Sumdata (ID%, UPDOWN%, TIMIDX%, SPEED%, LTOT!)
1305:
1306: SPDDAT!(ID%, UPDOWN%, TIMIDX%) = SPDDAT!(ID%, UPDOWN%, TIMIDX%) + CSNG(SPEED%)
1307: DTCNT%(ID%, UPDOWN%, TIMIDX%) = DTCNT%(ID%, UPDOWN%, TIMIDX%) + 1
1308: JAMDAT!(ID%, UPDOWN%, TIMIDX%) = JAMDAT!(ID%, UPDOWN%, TIMIDX%) + LTOT!
1309:
1310: END SUB
1311:
1312: SUB Sumdata2 (INS$, SD(), ND(), JAM AS JamF)
1313:     CALL BARA(INS$, JAM)
1314:     I% = VAL(JAM.DIRECT)
1315:     SD(I%, 1) = SD(I%, 1) + VAL(JAM.TRAF.ALL): ND(I%, 1) = ND(I%, 1) + 1
1316:     SD(I%, 2) = SD(I%, 2) + VAL(JAM.TRAF.BIG): ND(I%, 2) = ND(I%, 2) + 1
1317:     TT = VAL(JAM.TRAVEL.HH) * 3600 + VAL(JAM.TRAVEL.MM) * 60 + VAL(JAM.TRAV
EL. SS)
1318:     SD(I%, 3) = SD(I%, 3) + TT: ND(I%, 3) = ND(I%, 3) + 1
1319:     SD(I%, 4) = SD(I%, 4) + VAL(JAM.STOPCNT): ND(I%, 4) = ND(I%, 4) +
1
1320:     SD(I%, 5) = SD(I%, 5) + VAL(JAM.SPEED): ND(I%, 5) = ND(I%, 5) + 1
1321:     SD(I%, 6) = SD(I%, 6) + VAL(JAM.ACCEL): ND(I%, 6) = ND(I%, 6) + 1
1322:     SD(I%, 7) = SD(I%, 7) + VAL(JAM.AACCEL): ND(I%, 7) = ND(I%, 7) + 1
1323:     SD(I%, 8) = SD(I%, 8) + VAL(JAM.STOPTIM): ND(I%, 8) = ND(I%, 8) + 1
1324:     J% = VAL(JAM.JAMLEN1) + VAL(JAM.JAMLEN2)
1325:     IF J% <> 0 THEN
1326:         SD(I%, 9) = SD(I%, 9) + VAL(JAM.JAMLEN1): ND(I%, 9) = ND(I%, 9)
+ 1
1327:         SD(I%, 10) = SD(I%, 10) + VAL(JAM.JAMLEN2): ND(I%, 10) = ND(I%,
10) + 1
1328:         SD(I%, 11) = SD(I%, 11) + VAL(JAM.JAMTIM1): ND(I%, 11) = ND(I%,
11) + 1
1329:         SD(I%, 12) = SD(I%, 12) + VAL(JAM.JAMTIM2): ND(I%, 12) = ND(I%,
12) + 1
1330:         SD(I%, 13) = SD(I%, 13) + VAL(JAM.JAMSPED): ND(I%, 13) = ND(I%,
13) + 1
1331:     END IF
1332:
1333: END SUB
1334:

```

6. 旅行速度常時観測調査 実施要領

6. 1 目 的

本調査は、旅行速度の経時変化を把握することを、目的とする。

時々刻々と変化する交通状況を調査しその路線の交通特性を把握するには、交通量の他に旅行速度や渋滞長等も重要な情報であり、かつ経時変化を捉えることが望まれる。特に、都市部の路線では渋滞対策を効果的なものとする観点からもこれらの状況をモニタリングする意義は大きい。このため、旅行速度調査を常時観測調査として体系化し、道路の整備効果や交通状況を把握する必要がある。

道路の交通特性を表す指標として、交通量が多く調査され道路を計画する場合重要な指標となっているが、道路利用者にとっては、交通量よりもむしろ目的地までかかる移動時間が問題である。移動時間を適切に表す指標として旅行速度があるが、旅行速度は時間的に変化しその変化について細かく調査したデータは少ない。

本調査は道路交通センサスで実施した路線すべてについての調査ではなく、混雑区間の1部を調査の対象区間として、変動する旅行速度の経時的なデータを数多く収集し、交通量常時観測データと組み合わせて解析しその特性を把握する。また速度変化、渋滞長等とのデータとあわせて快適性、信頼性、定時性等の指標作りのための資料とするものである。

本調査要領は、現在観測機器が各事務所に配備されていないことや経時的な旅行速度データの収集は労力を要することを踏まえ、今後1～2年の暫定的調査方法を示したものである。今回の調査により、データの収集頻度、指標について解析したのち、旅行速度の常時観測調査を体系化する予定である。

6.2 調査手法

調査は、旅行速度測定装置一式を取り付けた試験車両を用いて行う。

従来の試験車を走行させて調査員がストップウォッチ等により計測する手法では、走行中に数多くのデータを測定することは煩雑である。そこで、試験車に旅行速度測定装置を取り付けて調査し、データを蓄積する方法をとることによって本調査の省力化と多様化を図る。

走行データとしては、一定の距離毎に所要時間、累積走行距離、渋滞原因等が簡単なボタン操作で内蔵のメモリーに記録され、更にそのデータをフロッピーディスクに収録し、走行状態解析プログラムを用いることによって旅行速度の経時変化、渋滞状況の実態等を把握するものである。

6.3 調査区間

調査区間は渋滞が問題となっている区間とする。

調査区間は旅行速度測定装置を保有する管内直轄路線から抽出するものとする。

また、調査対象区間は下記の条件を参考に設定する。

- ・交通量が多く、慢性的な渋滞が発生している区間。
- ・交通量常時観測地点がその調査区間内、もしくは調査区間付近に在り、その路線の時間交通量（方向別）が把握できる区間。
- ・測定延長が短いと信号の影響を受けやすいため、長い方が望ましいが、測定間隔の都合上混雑時でも1時間で往復可能な距離の区間。
- ・測定区間の延長は、概ね5km～10kmとする。
- ・調査区間は連続したセンサス区間（最小は1センサス区間）とする。
- ・改築、アクションプログラム計画地点等の区間が含まれているのが望ましい。

6. 4 調 査 日 時

旅行速度の季節変動、週変動、時間変動の状況を測定できる様に調査日を設定する。

四季観測の時期は原則として、春は4月、夏は7月、秋は10月、冬は1月とするが、その季節を代表する交通状況が把握できる日を設定する。

具体的には次の様に設定する

平成3年10月 1週間・・・・・・・・・・7日

平成4年1月 火水木のうち1日、日曜日1日・・2日

平成4年4月 火水木のうち1日、日曜日1日・・2日

平成4年7月 火水木のうち1日、日曜日1日・・2日

6. 5 調査対象時間帯

7時から19時までの各時間帯ごと毎12時間の旅行速度を観測する。

原則として、測定データは、各毎正時から次の毎正時までの間に観測したものを各正時の旅行速度として取り扱う。

旅行速度の観測は、各時間帯ごと1時間内に上り、下りの旅行速度データを測定する。したがって、1日あたりの観測は、観測区間を12往復することになり、1日あたりの旅行速度のデータは12回×2方向＝24個のデータを測定することになる。

なお、混雑時間が7時から19時内に納まらない場合には、この時間を越えて観測するのが望ましい。

6. 6 試験車両の運行計画

調査区間を定められた時間に効率的に観測が実施できるように、試験車両の運行計画を立てる。

運行計画の留意点

- ・ 時間調整をするための待機場所、回転場等の確保。
- ・ 試験車両は、乗用車又はライトバンとし、排気量が1500～2000cc程度の乗用車相当とする。
- ・ 試験車両の運転者の安全に対する配慮、交代等、休憩時間は運行計画で十分ねておく。

6. 7 走行方法

走行方法は、その道路交通の「流れに沿った」走行とする。

試験車両と同じ車種による追越し、被追越しが起こらないように、又は追越し、被追越しの回数がほぼ等しくなるように平均的走行を行う。したがって大型貨物車、バス等が低速度で走行している場合は適宜追越ししてもよい。4車線以上の道路でも同様であり、常に走行車線を通行する必要はない。

6. 8 データの集計

走行データは、走行解析プログラムを用いて、解析及び集計を行う。

6. 9 機器操作要領

概 略

・調査員（運転手）の走行中の操作は、調査起点にさしかかった位置でリモートボックスの「積算スタート」スイッチを押し、調査終点で「積算ストップ」スイッチを押し。この間に、自然渋滞（信号交差点等を含む）以外の理由での渋滞（事故、工事、その他）があった場合にはその都度、原因が判明した時点で該当のマーカーボタンを押す。

なお、センサス区間の判別、或は信号交差点の位置等は、別途作成する路線データを用いることによって解析プログラムで自動的に抽出される。

操作手順

（１）初期設定及び確認事項

① 電源投入

本体部の電源SWにて電源を入れる。

電源を投入すると自動的に装置の動作チェックがなされ、チェック終了後、本体部のモニタの表示が試験モードとなる。

②日時確認

本体部のモニタ表示モード切換えにて年月日、時分秒の確認する。

本体部の表示モード切換えボタンを押し、Calender & Date（表示モード5）を表示させて確認をとる。正しければ③の操作は省略できる。

③日時設定

本体部のカレンダー・タイマ設定にて年月日、時分秒を合わせる。

年月日／時分秒ボタンを押して年月日に合わせる（年月日側ランプ点灯）。年月日を設定したらSETボタンを押す。次に、同様に時分秒に合わせて（時分秒側ランプ点灯）、時分秒を設定し、SETボタンを押す。

④サンプリング間隔設定

本体部のサンプリング間隔設定スイッチにて、設定間隔「05」を設定、確認する。

データを取り込む間隔÷20cmからサンプリング間隔の設定値が決まるが、原則として1m間隔でデータを取り込むため、サンプリング間隔は「05」となる。

なお1m間隔より短い距離間隔でデータを取り込む場合は車両の最高速度が制限されるため通常は使用しない。

⑤モニタ確認

本体モニタ部の表示が試験モード（表示モード1）になっているかどうかの確認、設定を行う。

試験モードにしておくことによって、走行調査時の走行速度、積算距離、積算時間がモニタ可能となる。

⑥新しいフロッピーディスクの初期化

新しいフロッピーディスクを装置に入れ、初期化ボタンを押す。初期化が終了するとドライブ装置の稼働赤ランプが消え、モニタ部にOKと表示される。終了したらフロッピーディスクを抜いておく。この操作は必ず車両が停止した状態で行う。

調査終了後のデータセーブ直前でも、初期化は可能であるが、所要時間あるいは誤操作防止（既にデータが格納されているフロッピーディスクを初期化してしまう誤操作）の観点から、調査開始前に必要枚数を初期化しておく。この操作は走行中に行ってはならない。また、フロッピーディスクを挿入したまま走行することは絶対に避ける。

⑦車両進行方向表示ランプの確認

本体部の車両進行方向ランプ「前進」「後退」の表示（ランプ点灯）が、車両走行方向と一致するかどうかを車両を走行させて確認する。表示が逆の場合は、切り換えスイッチにて切り換える。

回転検出器の車両取り付け状態は車種によって異なるため、検出器の取り付け後、車両を前進（または後退）させて本体部の進行方向表示ランプの正逆を確認し、逆の場合は本体部にある切り換えスイッチを切り換える必要がある。

なお、調査期間中同一車両を使用する場合、この確認操作は最初に一度行えば良い。

(2) 調査時操作

①コメント№設定

リモートボックスのコメント№設定スイッチで、調査する路線の番号を設定する。

例えば調査路線が国道14号上りなら0141、下りなら0142と設定する。

②調査開始

リモートボックスの積算スタートボタンを押す。

車両が調査開始地点に達したら積算スタートボタンを押す。

③渋滞原因について

信号待ち以外の渋滞（事故、工事、その他）については、渋滞原因が判明した時点で所定のマーカを押す。

今回の調査におけるマーカの割付は以下の通りとする。

「事 故」：事故渋滞

「工 事」：工事渋滞

「幅員狭小」：その他

なお、信号交差点を先頭にした自然渋滞は、すべて信号待ちによるものとし、マーカを押す必要はない。

④調査終了

リモートボックスの積算ストップボタンを押す。これで1回の調査が終了となる。

車両が調査終了地点に達したら積算ストップボタンを押す。

⑤調査の繰り返し

1つの調査区間毎に①から④の操作を繰り返す。ただし、メモリ上に一時記憶できるのは合計50km (50,000ステップ) までである。

1回の調査終了毎にフロッピーディスクにデータをセーブしなくとも、メモリ容量あるいは10回まで調査を繰り返すことが出来るが、可能な限り1～2回の調査終了毎にフロッピーディスクにセーブするのが望ましい。

⑥データの保存

当日の調査を終了する場合、またはメモリに記憶してきた調査路線長に次の調査予定路線長を加えると50km (50,000ステップ) を越える場合は、メモリ上のデータをフロッピーディスクに保存する。

車両を停止させた後、予め初期化したフロッピーディスクを装置にいれセーブボタンを押す。セーブが終了するとドライブ装置の稼働赤ランプが消え、モニタ部にOKと一時的に表示される。

まず、駐停車可能な場所まで車両を移動して停止した後、所定の操作を行う。セーブが正常に終了した時点でメモリのデータが消去される。

使用できるフロッピーディスクの規格は3.5インチ2HDである。

セーブ終了後、改めて調査を開始する。

フロッピーディスクを装置にいれたまま、車両を走行したり、調査を行わないこと。

(3) 調査終了後

フロッピーディスクが装置に入っていないか確認してから電源を切る。

(4) その他

メモリクリア

走行テスト等で、フロッピーディスクにセーブする必要のないとき、メモリクリアボタンを押すと、それまで調査したデータを消すことができる

調査記録簿

路線番号

観測沿線市

観測日 平成 年 月 日 曜

観測 時間	方向	出 発 時 刻	天 候	視 界	路面 状況	工 事	事 故	備 考 その他	運 転 者	709t イ-番号	全交通 量	大型車 交通量
7:00	上り	:										
	下り	:										
8:00	上り	:										
	下り	:										
9:00	上り	:										
	下り	:										
10:00	上り	:										
	下り	:										
11:00	上り	:										
	下り	:										
12:00	上り	:										
	下り	:										
13:00	上り	:										
	下り	:										
14:00	上り	:										
	下り	:										
15:00	上り	:										
	下り	:										
16:00	上り	:										
	下り	:										
17:00	上り	:										
	下り	:										
18:00	上り	:										
	下り	:										

天 候

視 界

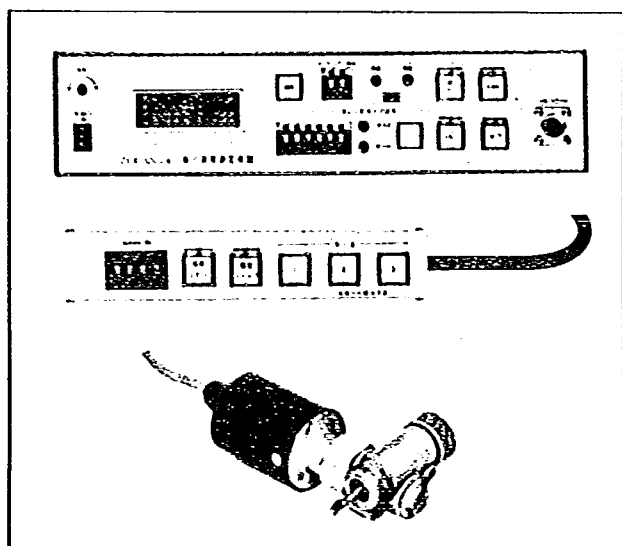
路 面 状 況

晴 1
曇 2
霧 3
小雨 4
雨 5
雪 6

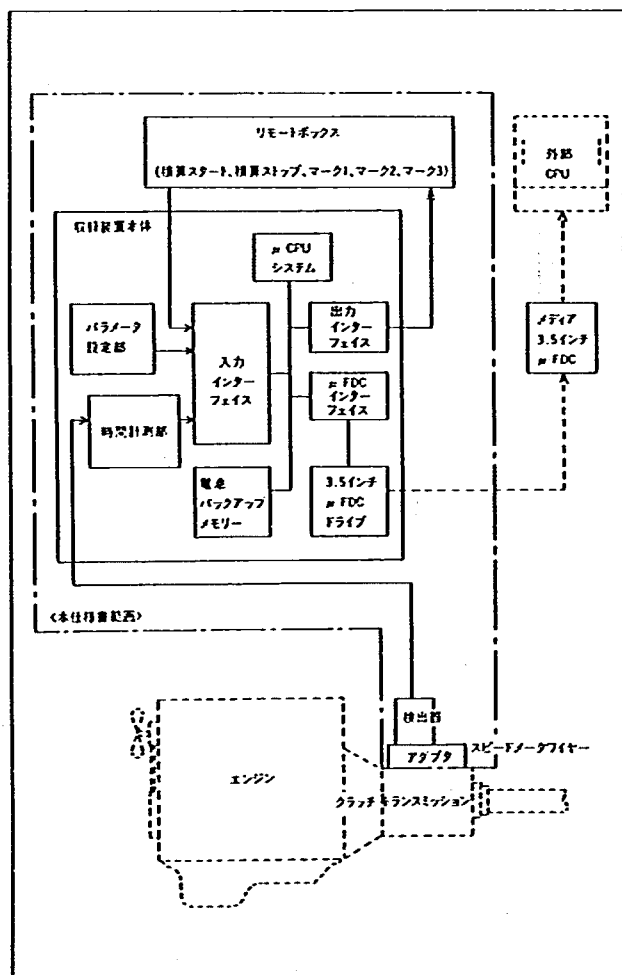
良好 1
夕暮れ 2
(スローラング点灯)
暗がり 3
(ハットライト点灯)

乾燥路面 1
湿潤路面 2
新雪路面 3
圧雪路面 4
凍結路面 5

6. 旅行速度測定装置の概要



概略ブロック図



【概 要】

本装置は、車輛の旅行速度をある一定距離単位で計測するとともに道路状況をマーカ信号として同時記録することにより、事後処理時においても交通状態を加味した分析が可能です。車輛のトランスミッション終段に回転検出器を取付け、その出力パルスから一定距離当りの所要時間を計測し車輛の旅行速度を算出するとともに記録が可能な旅行速度測定装置です。

【特 長】

従来、一般的に用いられていた方式として、実車の速度を計測しデータをデータレコーダに記録し、後に再生器を介してCPUに取り込む方式がありましたが、本方式には以下の長所があります。

- ①一定の移動距離毎にデータを収録する方式のため無駄なデータを収録することがない。
- ②①の理由及び、記憶媒体に半導体メモリーを使用するため、必要なだけ連続で収録することが出来る。（メモリー容量の増設は可能）
- ③記憶媒体に半導体メモリーを使用するため、振動に強い。
- ④電池バックアップメモリーを使用するため、データは確実に保存される。
- ⑤再生の必要がなく時間が省略出来る。
- ⑥計測の初段からデジタル計測でデジタル収録方式及び、時間計測方式のため、高精度が確保出来る。
- ⑦ランニングコストがかからない。
- ⑧外部CPU側のデータ取り込みには、最も一般的なOSであるMS-DOS上で、3.5インチマイクロフロッピーディスクをメディアとして使用できる。

【構 成】

- 検出器
- 収録部本体
- リモートスイッチボックス

【仕 様】

1) 検出部

使用車速範囲：0～160km/h
 使用温度範囲：0～40℃
 使用湿度範囲：80%RH以下(非結露)
 構造種別：JEM IP-54
 耐振性：5G(XYZ 3方向各2時間)
 耐衝撃性：50G(3回)
 正逆回転対応機能あり

2) 収録部

表示内容
 1) 実行ファイルNo
 2) 実時刻
 3) 積算走行距離

4) 積算所要時間

5) 走行速度

6) ファイルの使用状況

データ収録部

3.5インチ 2 HD マイクロフロッピーディスク
 MS-DOS

リモートボックス部

積算スタート/ストップスイッチ

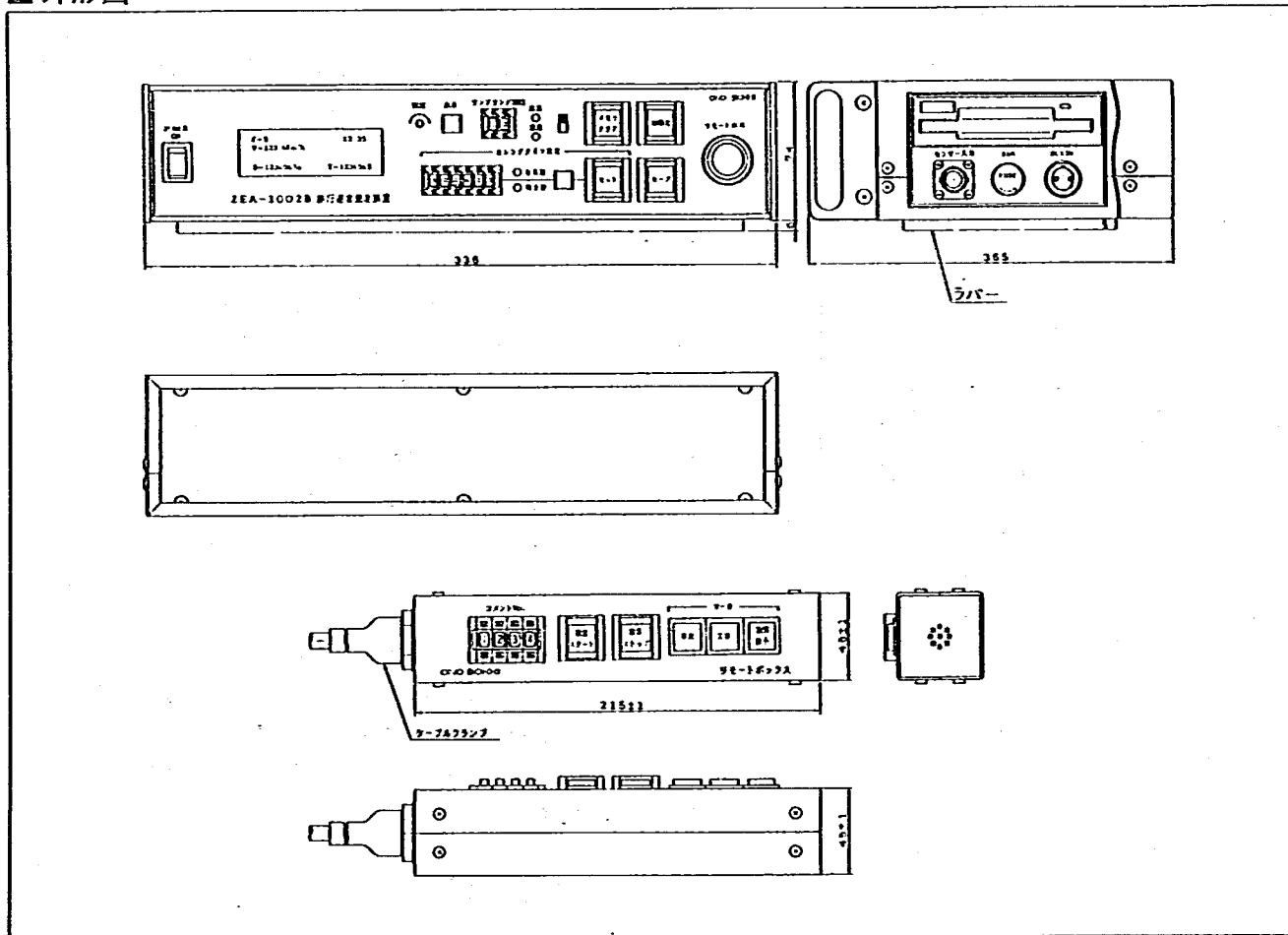
マーカー書き込みスイッチ

コメントNoスイッチ

電 源：DC12V(10V～18V)、約5A

塗装色：マンセル5Y7/1(小野測器標準塗装色)

■ 外形図



旅行速度集計解析プログラム
使用の手引

土木研究所資料第3071号
平成4年1月 (1992)
新交通研究室 室長 山田晴利
研究員 河野辰男
吉川 修

要 旨

道路の交通特性を把握するためには、旅行速度調査を常時体系的に実施する必要がある。また当研究室では、旅行速度調査の効率的な運用を図るため旅行速度測定装置の開発を進めてきている。本報告書は旅行速度測定装置を使用して旅行速度常時観測を実施するための、集計プログラムの使用の手引である。

キーワード：旅行速度調査、機械観測、渋滞調査、プログラム、使用説明書

郵便はがき

41円切手
をはいて
下さい

305-□□

建設省土木研究所

道路部新交通研究室 御中

茨城県つくば市旭一番地

No. 001

No. 001

配布先氏名

配布先住所



土木研究所資料第3071号

上記のとおり受領しました。

所 属(住所)

氏 名

印

土木研究所資料第3071号

