

マイコンによる天気予報

— グラフィックの活用 —

34期生

I テーマ設定の理由

いろいろな情報が飛び交う情報化時代の現在、コンピューターは、いろいろな所で情報処理に役立っている。このようにコンピューターは、多くの場面で活躍しているが、その反面、コンピューターを使いこなせない人も多い。コンピューターを導入しても、使わないで放置、ということになっている場合が結構多い。コンピューターも、人間が命令を下さなければ、ただの鉄とプラスチックとシリコンのかたまりである。このごろ、マイコンを使っている中学生をよく見かけるが、あの大部分は、雑誌にのっているプログラムを入れているだけなので、そういうことでは、本当に使いこなしているとはいえない。

僕達もそのようなコンピューターの使い方をしていたのだが、一度自作プログラムを作ってみよう、ということになり、テーマとしては生活に密着したもの、ということで、天気予報、という題材を取り上げてみた。

II 研究方法

1. 資料収集…マイコンの天気予報に関するもの

だが、そのような資料は、全くなく、結局僕達だけで独自にプログラムを開発することになってしまった。

2. プログラムの設定…フローチャートの作成

(フローチャート…プログラム制作のもととなるもののこと。)

3. プログラムの制作

1 地図表示部 4 等圧線表示部 7 天気予報&結果表示部

2 経線、緯線表示部 5 高低気圧DATA入力部

3 等圧線DATA入力部 6 高低気圧表示部

〈手順〉

1～7についてそれぞれについてプログラムを制作させ、それらをつなげて完成とする。

III 研究内容

1 地図表示部

写真1を見て欲しい。見たところ、かなり形が悪い無恰好な地図になっている。これは何故かと言うと、このコンピューター(NEC PC-8001)の分解能力の為である。この地図の場合、地図の線は小さな点をつなげて表現しているのだが、その点を打つ能力がこのコンピューター

の場合、縦 100 個、横 160 個となっているので、この様な地図となる。

さて、実際にプログラミングするのだが、今回は「タートルグラフィックス」という書き方で、プログラミングする事にした。多少、理解しにくいと思われるが、その文法を載せておく。

〈タートルグラフィックスの文法〉

I S E T (命令群…#)

- 命令の最後には、#を必ず入れる
- 変数は、1～2 文字の整数型で、命令中に現れる前に、初期値を代入すること
- 命令群について
 - e x p = 変数、定数 V = 変数
 - X e x p 画面上の X 座標を与える
 - Y e x p 画面上の Y 座標を与える
 - C e x p 出力カラーの指定
 - ± V 変数の値を増減させる
 - S 現在の位置に点を打つ
- 移動命令 e x p 引数は移動距離を与える
- 命令間には、#を入れる
(移動命令については、右の表を参照)
- 繰り返し命令
 - [e x p, …] [] で狭まれた間の命令を引数の回数だけ繰り返す
 - [e x p, + V] } 変数の増減
 - [e x p, - V] }

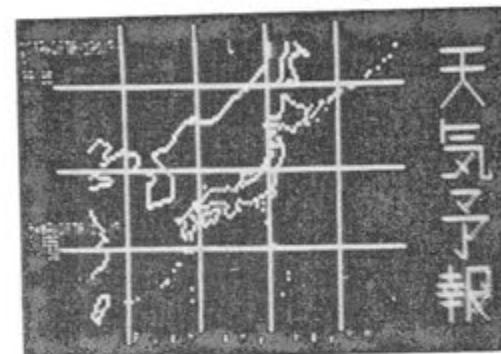
さて、これらの命令を実際に用いて、写真 1 のような日本とその付近の地図の表示を実際に行った訳だが、よくわからん、という人も多いと思う。そこで、ここでは実際に、台湾のプログラミングの手順と、その地図の L A Y O U T を載せてみることにした。



〈台湾の L A Y O U T 〉

それでは実際にプログラミングしてみよう。1 から点を打ち始め、左上に 2 つ、上に 1 つ、右上に 1 つ、上に 2 つ、右上に 1 つ、上に 1 つ、右上に 1 つ、右に 2 つ、下に 3 つ、左下に 2 つ、下に 1 つ、左下に 1 つ、下に 2 つ、ということになる。つまり、1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 10 → 11 → 12 → 13 → 14 → 15 → 16 → 17 → 18 → 19 → 20 と、打たれていくわけである。これをプログラムにすると、こうなる。

```
I S E T( )U2, U1, (U1, U2, (U1, U1, (U1, R2,  
D3, )D2, D1, )D1, D2, #)
```



〈写真 1 〉

跡を残す	跡を残さない	方向
U	J U	↑
D	J D	↓
R	J R	→
L	J L	←
(U)	J(U)	↗
(D)	J(D)	↘
(U)	J)U	↖
(D)	J)D	↙

〈移動命令表〉

これで、なんとなく理解できた、という人もいるのではないか、と思う。下に、全 L A Y O U T を載せておく。

PRINT LAYOUT FORM												
PROGRAM NAME	LAYOUT NAME		CAUTION		COORDINATED BY		MODE		NO.		PRINT LAYOUT FORM	
F300 0												
F378 1												
F4F0 2												
F468 3												
F4E0 4												
F558 5												
F5D0 6												
F548 7												
FAC0 8												
F7B0 9												
F828 10												
F8A0 11												
F8A0 12												
F818 13												
F830 14												
F808 15												
F8A0 16												
F8F0 17												
FB70 18												
F8C8 19												
FC60 20												
FC08 21												
FD50 22												
FD08 23												
FE40 24												

2 経線、緯線表示部

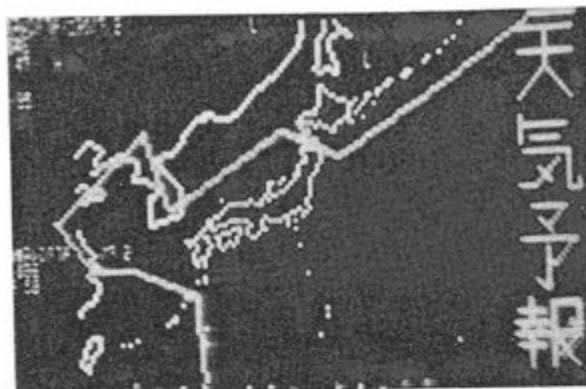
写真 1 を見て欲しい。地図上には経線、緯線がある。これには、L I N E 文、という命令が使われている。LINE 文というのは、画面上の 2 点の座標を与え、その間を点で結ぶ、というものである。右に、そのプログラムを載せておく。尚この緯線、経線はあまり多く引くと画面が見にくくなるので、10 度おきに引くことにした。

3、4 等圧線 DATA 入力部 & 表示部

最初、僕達二人は、テレビでやっている天気予報の等圧線の様に、何 mb かおきに等圧線を引こう、と考えていた。だが、とてもあれだけ線を引くのは、僕達の頭脳では難しく、また、あまり多く引くと、天気図が見にくくなってしまうことが分かった。

そこで僕達は、ラジオの気象通報で報じられている「日本付近を通る等圧線」の DATA を利用することにした。

ラジオで報じられる等圧線の通っている緯度、経度を、画面上の X、Y 座標に変換し、その間を先程使った L I N E 文で結ぶことにした。したがって、この等圧線には角がある。



が出ており、このことからも、夏の天気図であることが分かる。

5. 6 高低気圧DATA入力部&表示部

入力手順を書くと、まずDATAの個数を入力し、(緯度、経度で1 DATA)次にDATAの緯度、経度、を入力する。入力された緯度、経度、それらを、画面上の座標に変換して、LOCATE文と呼ばれる命令を用いてカソールを変換した座標に持っていく、そこに高気圧、低気圧、それぞれ'H'、'L'、と表示させる。この操作を、DATAの個数だけ繰り返す。上の写真にも、それらがディスプレイされている。

ここでは高気圧について
のプログラムを載せておく。
低気圧も同様なプログラムとなっている。割合簡単なプログラムになっている。

```

200 PRINT CHR$(T2)
210 INPUT "コウキアツDATA/コスウハ"; A1
220 FOR A4=A1 TO 0 STEP -1
230 INPUT A2, A3
240 LOCATE FIX(((A2-120)*3+20)/2), FIX(((50-A3)*3+5)/4), 1
250 PRINT "H"
260 NEXT

```

7. 8 天気予報部&結果出力部

ここが、この研究の心臓部なのだが、はっきり言うと僕達はこれをプログラム化することができなかったのである。つまり予報のできない天気予報になってしまった。という訳である。理論的な物については一応完成していた。どのような物かというと、ただ今現在の雨域のDATAから、予報したい時の雨域を出して、そこに予報したい所は入っているか、否か、知ろう、というものである。

これをプログラム化できなかった理由としては
。そこまで予報するだけのDATAが手に入らない。
。僕達の頭脳が未熟である。
などが挙げられると思う。

いずれにしても、この研究は大失敗なのである。一応、結果出力部はプログラム化できたので書きたいと思う。予報できなくても、結果は出るのである。

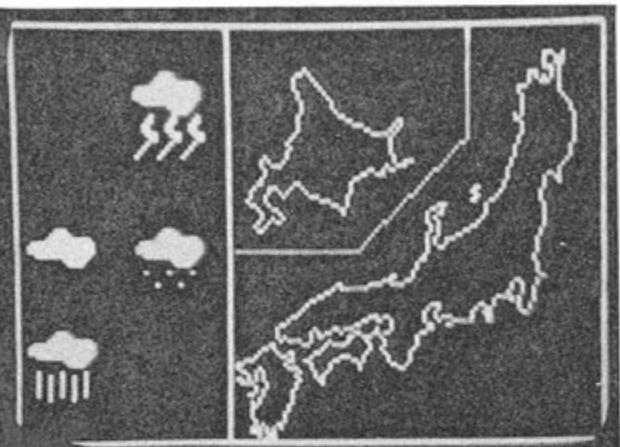
〈写真の説明〉

7月下旬のある日の天気図である。1008mbの等圧線が1本ある。高低気圧の表示もされており、それぞれH, Lで表示されている。右の方の「天気予報」の文字は、タートルグラフィックスであり、この天気図も実際の画面上へは、カラーでディスプレイされている。等圧線に夏型の特長の、「鯨の尾」

表示方法としては、ここでは一つタートルグラフィックスを用いて、天気予報表示用の画面を作ることにした。

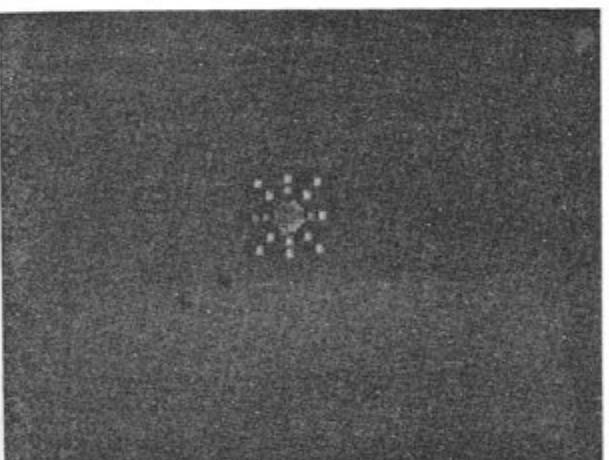
右の写真がそうである。書き方としては、地図の枠はLINE文、その他はタートルグラフィックスを使用した。この写真ではあまりよくは分からぬと思うが、実際には、晴れは赤、曇は白、雨は水色、雷は黄色、というふうに、配色にも工夫してみた。

さて、予報結果の表示だが、最初は、右の写真の地図上に表示しよう、と考えていた。だが実際にやってみると、地図の様子が、雑



〈あすの天気は〉

〈結果表示〉



然としてくるので、一度画面をクリアして、そこにマークのみ表示することにした。左の写真がそうである。先程のマークをそのままにしたので、すこし間の抜けた感じの表示になってしまった。

結局僕達は、予報については何もできなかつたので、この様なことをしても意味はないのだが、後に予報プログラムが完成すれば、このプログラムも無駄にならないで済む、と思っている。

IV 結論

コンピューターは、命令しないと何もしてはくれない。つまりプログラムしなければいけないのだが、そのプログラムが、僕達の場合には完全に完成することはできなかつた。グラフィック部分は、大部分完成したが、予報部分、つまりコンピューターに判断させる部分については、プログラムにできなかつた。これから分かったのだが、画像を描く、というのは、画面上の座標に画像を置き換える、命令する、ということは易しいが、判断させる、というのはプログラム化するのが難しいのである。プログラムの世界は、数学の世界である。だから、「判断」という心理的な物を、「数学」という数量的な物へ置き換えるのが難しい、ということにもなると思う。

V 轉 插

やはり、テーマが僕達には多少難しすぎた、といえると思う。相手の家に泊まりこんだりして、頑張ったが、結局うまくはいかなかった。あまりにうまくいかなかつたので、気象庁のコンピューターのプログラムはどうなってるのか?などと、思ったりもした。

失敗したのに何故ここに載れるのか、と考えてみると、やはり今迄に、マイコンについて研究した人はいても、実際にプログラムをたててみた人があまりいなかったのではないか、ということである。コンピューターによるディスプレイということも、みんなの興味を引いたのかもしれない。

やはり、コンピューターを使いこなすのは、大変難しいことだ、と思うが、これからコンピューターの時代がやってくるのは目に見えている。難しくても使いこなさなくてはいけないのである。間違ってもコンピューターに振り回されて使われる人間になってはいけないと、僕達は考えている。

〈今回僕達が作ったプログラム〉