LiMe

'Limited Expression' reports

0010 目次 28 54 52 55 45 20 49 46 20 43 48 41 52

絵・印刷 H. Fukushima

01D0 20 45 51 55 20 20 20 20 20 31 35 20 20 20 20 20

BOX用


Toeidoga

発刊に当って

部長　吉田　[illegible]

11月の声を聞くと毎年の様に学園祭に小冊子を出そうと言う声が聞かれる。だがかって今までにまともにそれが実現した事は一度も無かった。ところが今ここに実現をしたのである。

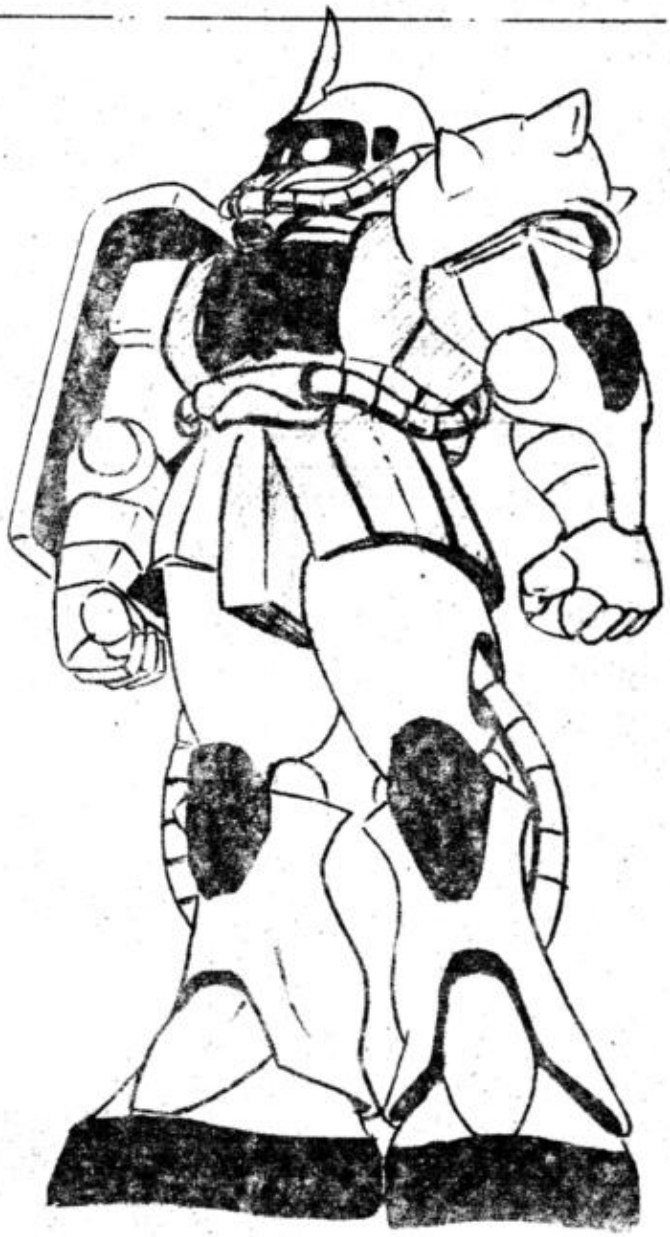
この2年私たちの部は大きく変った。わずか10名たらずであったそのが、昨年20名、今年は10名を越す入部者を迎え、大きくふくらんでしまった。今や30名を部員数は越すのである。それまではみんなが部室に集まって色々な事を話し会う、それで良かった。そしてそれが何の方向付けも無く大きくなってしまった。今私たちの部にとって一番大切な時ではないだろうか。それは1つの受精卵が2、4、8、16‥‥と未分化で分裂をくりかえして細胞を増やして行く、そんな過程に有る様に思う。さらに細胞は分裂をくりかえし、分化をして各器官へと成長し、生命が生まれる。私たちも何かの目的を見つけ、各々に分化し、そしてそれらが組み合わされて働いてこそ本当の部とな

るはずである。何の方向もなしにただ大きくなって行く、それではま[illegible]くのガン細胞だ。私はその何かをさぐりようと考えて来た。しかしまるっきりの闇で、手さぐりをしても何の感触も無かった。みんなそうであったと思う。この冊子にみんなの思う事や、4月からの活動の足跡を書く事で自分たちの進んで来た道や、これから進もうとする世界がおぼろげにでも見えて来るのではないだろうか。たとえホタルの光ほどにしかならなくとも、真暗よりはましなはずである。そして又、部外の人々に読んでもらう事により、私たちとは異なった次元から見た様子（あたかも航空写真の様に）を御意見としていただけるのではないかと思う。そして一刻も早く、はっきりとではなくともおぼろげにでも目標を見つけようではないか。

ただワヤワヤッとした”かたまり”になってしまうのか、それとも生命が生まれるのか。それは私たちに課せられた、どうしても答えを出さなくてはならない大命題であり、天が与えた試

練ではないだろうか。その解決への第一歩としてのこの小冊子が、1号だけで終らずに2号、3号と号数を重ねて行ける事を心から祈るものである。

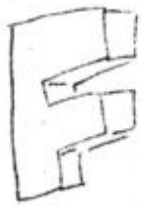
昭和56年11月



埋めくさげんこー #1.

書いた人：もちろん Editor Take.

かの米国には、Fortran-Manが居られるらしい。



Fortran-Manの似顔絵

もちろん、ALGOL-ManやPASCAL-Manは大悪人だそうだ。危機には、IBMの歌をうたって、うそぶくらしい。

これは、確か、Dittoの載っていたので知ってた人は、笑って下さい。知らなんだ人も笑え。

Fortranも77になって、ガムバッテいる。どうせなら、Local Variable導入して、再帰呼出し許せば、ALGOL-60ぐらいには、なるだろうて。ついでにストリング変数とストリング演算いれたら、Microsoftの BASIC程度になるのになあ。しかし、それ以上機能をつけたら、全然身動きのとれない肥満のお兄さん PL/1 みたいに、なっちまう。PL/1みたいに、みすぼらしくなったらおしまいだ。

なんでもできる言語なんか、ナンセンスだ。目的に合った言語を目的に合った様に使うから、良いパフォーマンスが得られるのだ。どんなことしても、十分の一程の機能しか使わないなら、十分の一の大きさの言語を十個持つ方が有利だ。

人工知能!?とは大ぜさな.

ーオセロべったり

コンピュータを与えられて、「さあ、何かやりなさい」とをおれたら、あなたは、何をしますか？
ボクラブもコンピュータ部なんて、ごたいそうな名前を頂いているものですから、何かしなければ、ならんのですが、これが意外にむつかしい問題なんですな。

一般に、コンピュータが利用されている方面としては、主に数値解析、データベースなどが有名ですが、数値解析といっても、先ず、何か研究対象があり、その解析にコンピュータを利用するのですから、その対象がなくては話になりません。又、データベースも、ソフトのテクニックなどを学ぶのなら、多くの蓄積があり、良き学習対象であるのですが、でき上がったSoftを使う場がなくては、作る気にはなれないでしょう。

つまり、コンピュータを使う事が第一義では、普通、何もできない訳ですな。

ところがですな、人工知能という分野が残っていたんですわ。人工知能とは、人間に見られるような知的な行動をコンピュータにやらせようというものです。この分野はコンピュータに

知的な行動をとらせる事。その一般的手法を発見する事が主たる目的で、従（実現させるための具体例）については、ゲームの相手をする。迷路を解く。証明する・パターン認識・PERTなどが有名ですが、その内容は様々です。

これなら目的のない人間でもできそうだ、（特にゲームの相手をさせるのは、それ自体がおもしろい。）と言うので「オセロの相手をする」コンピュータを作ろうというのが今回オセロをDEMOする大義名分でありましょう。

しかしですな、これが本当に作りだすと、えらい話が違ってくる訳ですわ。一般的手法として使えるようなものを見つけるなどという命題はどこへやら、「強くするにはどうすればよいか」これが主となり、その作業のほとんどは、オセロの解析という風になりますねん。（こういう傾向は、人工知能の開発で最も有名なChessをするプログラムに於てさえ、見られる事も本当なんですが、...）

という訳で、新たな一般的手法は見つける事ができていないのですが、（しかたがないので）オセロの思考ルーチンについて、以後、幾つか説明したいと、思います。

1. 下準備

いくら"強いオセロプログラムを作るんだ!!"と言っても、人工知能の基礎的手法位は、やはり使わなければ損というものでしょう。そこでわれわれ、代表的なMin-Max法などについて少し、説明します。

○木探索

コンピュータ上でオセロの局面を実現し、考える時、用いる一般的手法。ある局面から次の局面に移る時、その着手を枝局面を節と考えると、図1のように木のような形でゲームについて考える事ができます。

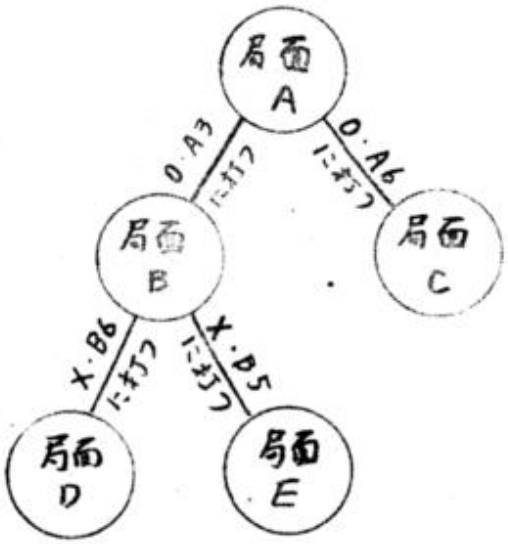


図1

○見えない木

オセロの必勝プログラムを作るとすれば、全ての手について考えればよい訳ですが、実際には、全ての手を考えるというのは不可能です。そこで局面の生成に制限を与えます。これを打ち切り条件といいます。

最初の局面、新たな局面を生成する規則、そして打ち切り条件を見えない木とまいます。つまり、見えない木が与えられると、各局面を生成できる訳です。

○ 静的評価関数

新たな局面を生成する事なく、その局面に評価値を与える関数。自分に有利な局面ほど高い値を示すようにな,ていることが望まれる。

○ 打ち切り条件

見えない木を与える際、全ての着手について考えることができないのだから、できるだけ有効な枝を生成するようにしなければならない。そこでどの枝を生成し、どの枝を刈るかを示すのが打ち切り条件である。

例えば「ゲーム終了」というのは絶対的な打ち切り条件であり、他に「最大深さ」「先枝刈り」「静かな局面」などという手法が用いられる。

○ 最大深さ

打ち切り条件として枝の深さを決めておくもの。いわゆる3手読みなどと呼ばれるのがこれである。

○ 先枝刈り

新たな局面を生成する際、探索しても、よい値が返ってきそうにもない枝について、あらかじめ刈ってしまおうというもの。(ただしこれには、大切な枝を、よく調べもせずに刈ってしまう危険がある)

先枝刈りの方法としては、

1. その局面において、各枝を評価し、上位n個の枝についてのみ探索をつづける。即ち、静的評価で無駄と思われる枝を刈る。
2. 上記において、レベルが深くなるにつれて、nの値を小さくする。
3. その局面を評価してみて、基準値以下になると、以後の局面は調べない。
4. 次の局面の評価値と、今の局面の評価値の差が基準値以下になると、以後の局面は調べない。

などがある。

○ 静かな局面

これは先枝刈りの4.とほぼ同じ効果を期待するものである。少し位探索したのでは評価値の変わりそうにない局面を刈る、というもので、逆に言えば、チェスにおける、チェックやフォークのような静かでない局面を、深く探索したいという事です。

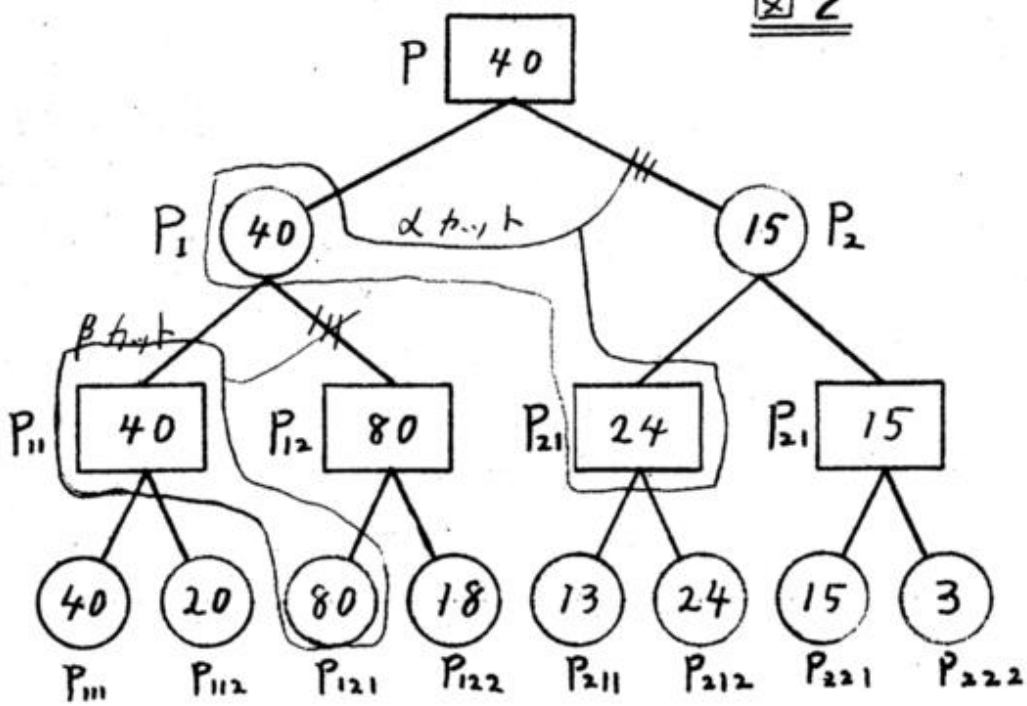
• ミニ-マックス法.

評価値を後続局面から繰り上げる方法の一つ。今、図2の様な局面が生成されていたとする。(一番下の局面の値は静的評価関数により、与えられているとする。) この時、P、P_1、P_{11} などに評価値を与える手法のこと。

コンピュータ側は自分に有利な局面を生成させようとするのだから評価値の高い枝を選び、同様に人間はコンピュータに不利な局面を生成させようと、評価値の低い枝を選ぶと考えられる。

例えば図2において P_{11} に評価値を与える時 P_{111} と P_{112} について考え、P_{11} はコンピュータ側の局面だから $Max(P_{111}, P_{112}) = P_{11}$ とする。(P_1 については $Min(P_{11}, P_{12}) = P_1$ となる。)

図2



又、ミニマックス法の修正版として、(呼称は知りません。すんまへん。)次の局面の上位n個について、平均を求めて、その局面の評価値とする。などというのもあります。

・α-β枝刈り

探索の始めに尽ては、当然ながら、局面Pだけが存在する。以後P_1やP_2を生成してゆくのであるが、$P \to P_1 \to P_{11} \to P_{111} \to P_{112}$のように局面を左側から順に生成してゆくのを縦型、$P \to P_1 \to P_2 \to P_{11} \to P_{12}$のように各レベルの局面を順に生成してゆくのを横型という。以後局面の生成は縦型として話をする。

今、局面P_2の値が決まったとする。すると、次にP_1の値を決めるべく、P_{21}の値を決めに行く。この時、P_{21}が24とわかると、$P_2 = Min(P_{21}, P_{22})$だから$P_2 \leq 24$となる。又、$P = Max(P_1, P_2)$で$P_1 = 40$だから$P \geq 40$とわかっている。従って以後の局面($P_{21}$など)は生成してもPの値は変わらない事がわかる。これがα枝刈りで、β枝刈りは、立場を入れ替えて、P_{11}が決定し、P_{12}を調べる時P_{121}がP_{11}をこえると、以後の局面(P_{122})は調べなくてよいというものである。

尚α-β枝刈りを多く起こさせるには、最初の方でよい手を生成することであり、そのためには

探索する枝の順序づけをするとよい。

この順序づけは静的評価だけでなく、例えば浅いレベルについて探索し(この場合、評価値は、最後まで枝を伸ばした時の値でない点に注意)その評価値が大きく違うようなら、探索順序を入れ替えるなどという手法もある。

以上が人工知能の一般的手法として、割合、有名なものである。

せやけどね、これらの手法がオセロにとって有効なものばかりとは言えませんねんな、中には大して効果を期待できんものもありまっせ。どの手法を選択するかが、強さを決める重要な手掛りとなりそうですなぁ。

2. 思考ルーチン

以後実際にプログラムする場合について考えます。

(a) 評価すべきだと思われる内容.

i) 位置による絶対評価、
ii) パス
iii) 確定石(以後決して返せない駒)の数
iv) 相手(又は自分)の返すことのできる駒の数
v) 〃 の打てる所の数
vi) 角

(b) 実現できそうな考え方

i) LAST

これは他の例とは違い、終盤についてのみの問題ですが、オセロは多くても60手で終わるとわかっていますから、残り8手位なら読み切ってしまおうという考えです。

時間が許すなら、これより強い手法はないので各プログラムに付加すべき手法でしょう.

ii) PATARN

(a) の i) のみに頼ろうという考えです。これは、実際には実現させても非常に弱いものです。特にこのプログラムの弱点は序盤であれ、終盤であれ一定の打ち方をする点です。

iii) PATARN'

そこで才何手目かによって Table の値をかえてやろうという訳です。しかしまだ、局面に無関係に打っただけです。

iv) PATARN''

(a) の vi) を少し考慮します。オセロの着手で特に重要なのは角とその まわり、及び辺ですから、角をとる、又はとられた場合、以後のテーブルを変更します。

◎ 以上 PATARN(''') については各場合の評価値は、結局人間が入力する事になりますので、Table の値をどのように決定するかで、強さが相当変化します。

v) STEADY

これも i) の LAST 同様、単独で動くソフトでは、ありませんが、a) の iv) について考えたものです。角をとると確定石が必ずできます。(角をとらなくても確定石はできうるのですが、これは識別

するのがむつかしいので実現の目途が全く立っていません。）そこで以後、確定石の数の差が最大になるように打とうという考えです。

vi) TOM

これはTitleからでは意味がつかめませんが、(a)のiv)を主に考えたものです。最後は駒の数の多い方が勝ちですので数多く駒を裏返せば有利だろうという訳です。元の局面に対しそのままこの考えを導入しますと、いわゆる初心者の打ち方で結果は歴然ですが、これを3手先、5手先の評価とすることなると話も変わりましょう。

vii) TOM′

前述のTOMでは角の評価をしていませんので、いくら先を読んでも、それだけではやはり弱いです。そこで生成した局面において角だけ特別な評価を与えようというものです。

viii) KOMA

これも名前だけでは意味がわかりませんが、(a)のv)について考えたものです。そもそもなぜ、相手の行ける所の数が評価の対象となるかというと、実は(a)のii)「パス」から派生したものです。理由はともあれ、相手に「パス」させると有利だとは、

直感的にも感じられると思います。又実際にもゲームの主導権が握れるなど利点があるようです。この「パス」というのは、つまり、相手の打てる所が零の状態と考えられます。そこで、相手の打てる所の数を最小にしようという訳です。

ix) KOMA'

KOMAでは相手の打てる所の数を評価値としましたが、これではどこに打っても同じ重みなので、いわゆるPATARN"のようなもので打てる所によって重みを変え、「相手の打てる所の各々における重みの和」を評価値としようというものです。

さっと示しただけでこのような考えがあるわけですが、どれもその手法のみでは、決して強いとは呼べず、序盤、中盤、終盤と、各局面に合った評価を用いる事が大切でしょう。

とはいうものの、各々の手法を実現するのさえ苦労している現状では、それらを統合した「強いオセロプログラム」はいつになったら、できることやら……はぁ。

END

自走型ロボット　制御班

[1] 概要

制御班では、コンピュータによる制御の対象物として、今回はロボットを取上げることにした。一口にロボットと言っても、多種多様であるが、次の様な特長を持つロボットを製作した。

(1) 自立型ロボットである。
ロボット自体が一台のコンピュータを有しており、他からの介入なく、作動できる。

(2) 黒い道を走る。又、前進、後進、右折、左折可能な為、交差した道を自由自在に走り回ることができる。

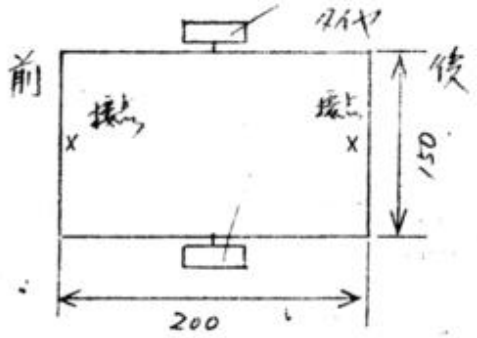
[2] ハード編

本体は底板となるフレーム部の上に電源部、駆動部等が乗っており、その上に、駆動系制御部及びセンス部用基板、CPU部用基板が固定されている。

(1) フレーム部
150×200 mm アルミニウム製 バネルチャンネル使用

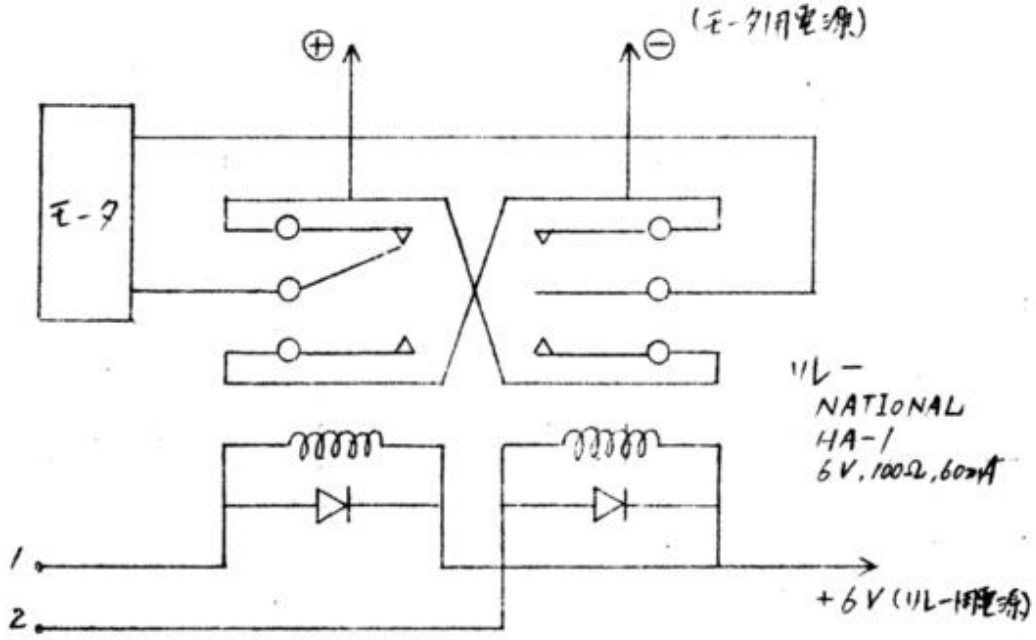
(2) 駆動部
2輪独立駆動、4接点、前後対称形 (Fig. 1 参照)
MABUCHI MOTOR RE 140 , TAMIYA HI-POWER GEAR BOX SET 使用

(3) 駆動系制御部
モータ制御は、トランジスタでも可能であるが、無難なところで、リレーを使用した。1モータに2つのリレーをクロス状に接続し、前進、後進、瞬時停止（逆起電力による）を可能にした。(Fig. 2 参照)



前後の接点は実情に応じて
ボールキャスタ等を使用

Fig 1. 底部



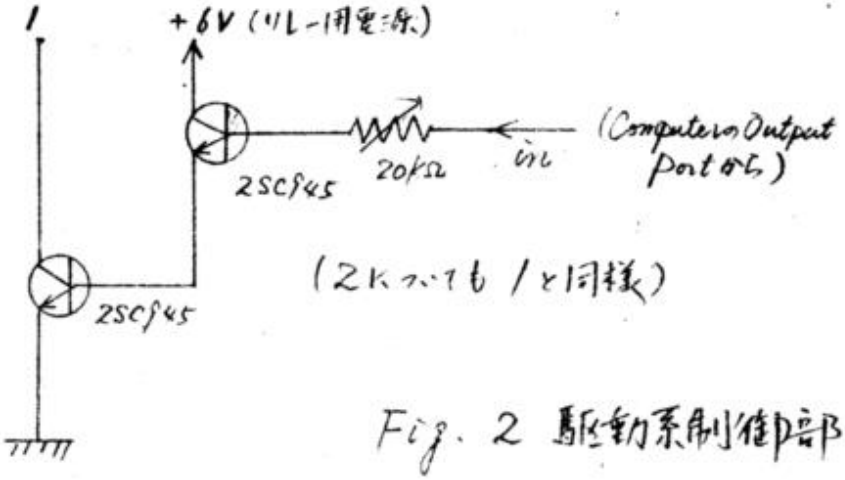


Fig. 2 駆動系制御部

(4) センサ部

黒い道識別の為、フォトトランジスタを使用、又、右折、左折、直進、後進等もフォトトランジスタの判定によって行なう。(Fig 3, Fig 4 参照)

(5) 障害物認識部

前後、2個ずつ、計4個の接触スイッチで障害物を認識する (Fig 5)

(6) CPU部

CPUには、MOTOROLA CMOS 8bit CPU MC 146805E2P使用 メモリは4k RAM (Fig 6)

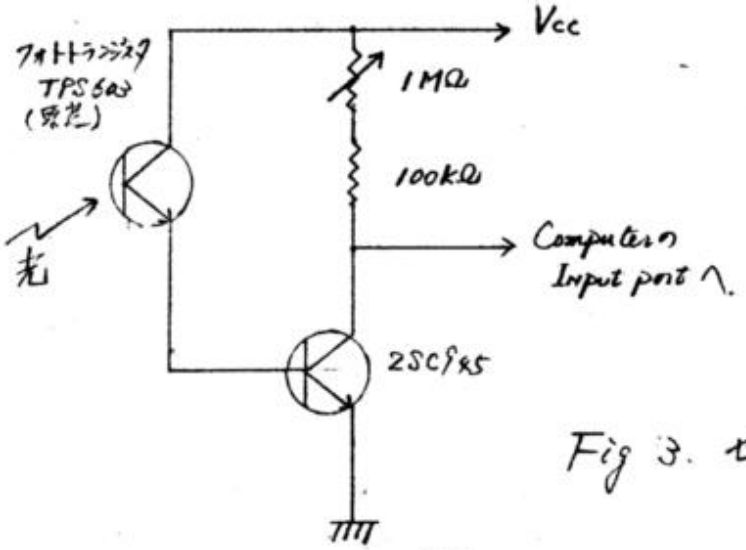


Fig 3. センサ回路図

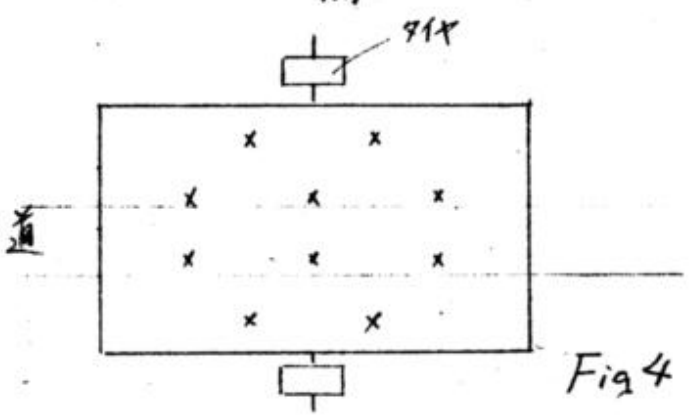


Fig 4 底部センサ配置図

(7) メモリ読み出し，書込み部

ブート・ストラッププログラム等はハードウェアパネルを用い，その他は，ポートを介してTK80-BSに継ぎ，ターミナルとして使用している

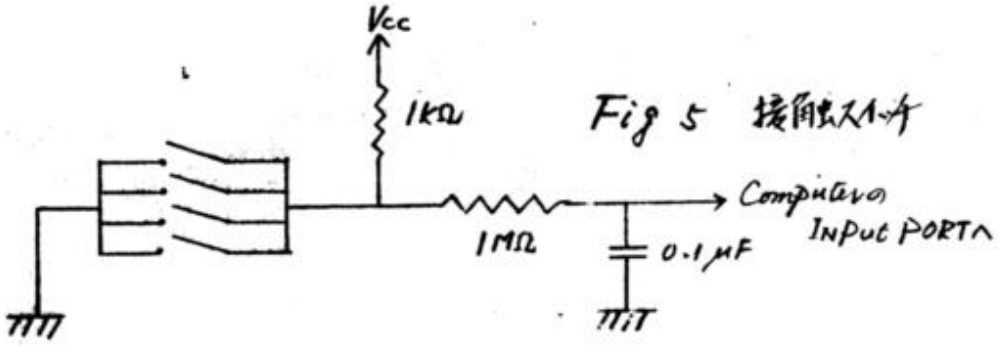
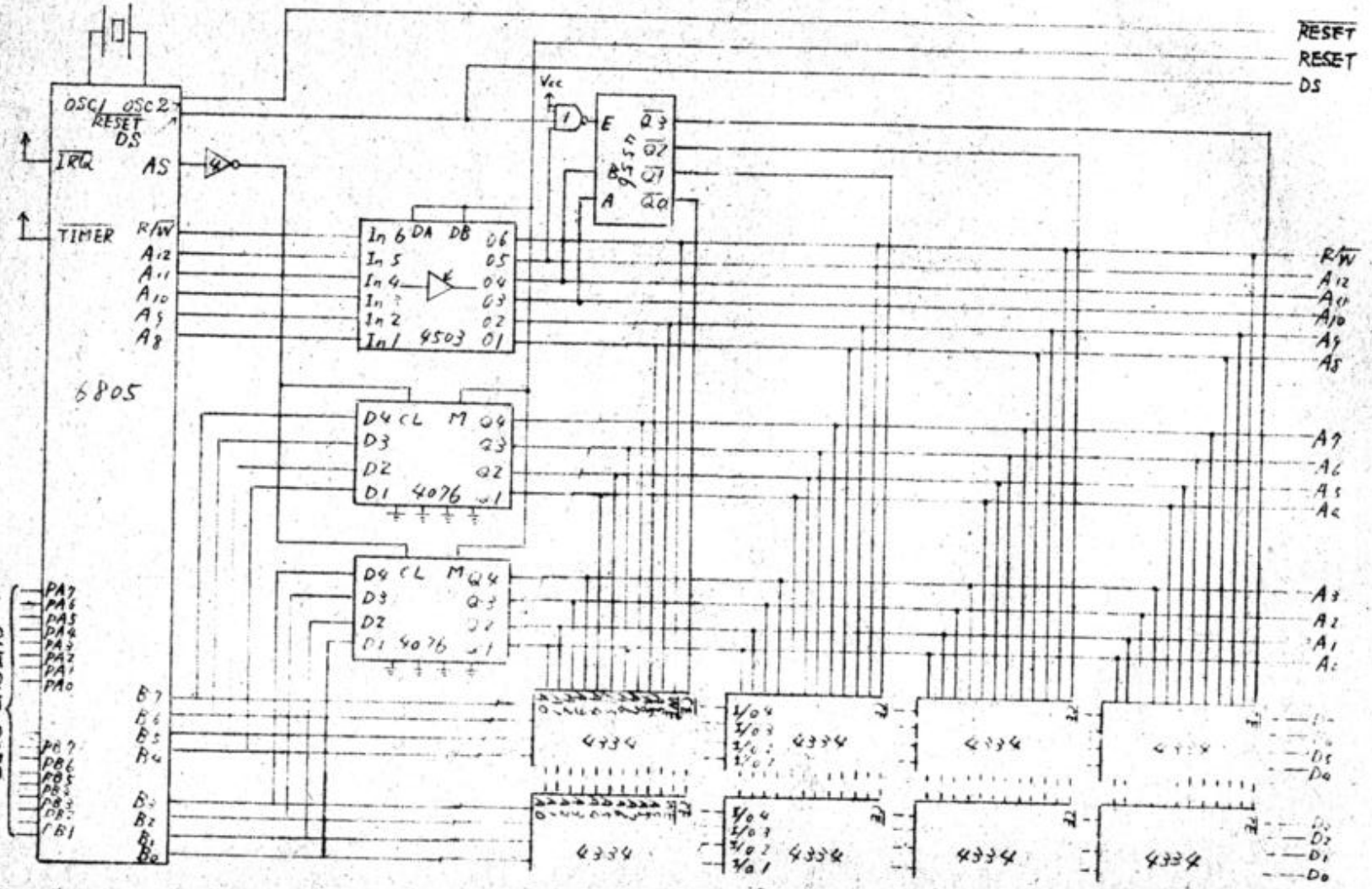


Fig 5 接触スイッチ

3 ソフト編

今回のロボット製作の第1目標は，交錯した道を自由自在に走らすことにある．人間でいうところの目である．センサを使用して，果してどこまで道を走ることが可能であるかが問題となる．そこで道を自由に走らすための以下の様な基本サブルーチンを作る．

(1) 進路修正サブルーチン
　　黒い道から車体がそれない様にする
(2) 直進サブルーチン
(3) 交差点検出サブルーチン
　　右折，左折，あるいは前進が可能か識別する
(4) 右折サブルーチン
(5) 左折サブルーチン
(6) 反転サブルーチン
　　前後の概念を逆にする
(7) 前進可不可識別サブルーチン


OSC1 OSC2
RESET
DS
IRQ
AS
TIMER
R/W
A12
A11
A10
A9
A8
6805
PA7
PA6
PA5
PA4
PA3
PA2
PA1
PA0
OUTPUT PORT
PB7
PB6
PB5
PB4
PB3
PB2
PB1
PB0
B7
B6
B5
B4
B3
B2
B1
B0
Vcc
E
A
B
4503
4076
4334
RESET
RESET
DS
R/W
A12
A11
A10
A9
A8
A7
A6
A5
A4
A3
A2
A1
A0
D7
D6
D5
D4
D3
D2
D1
D0

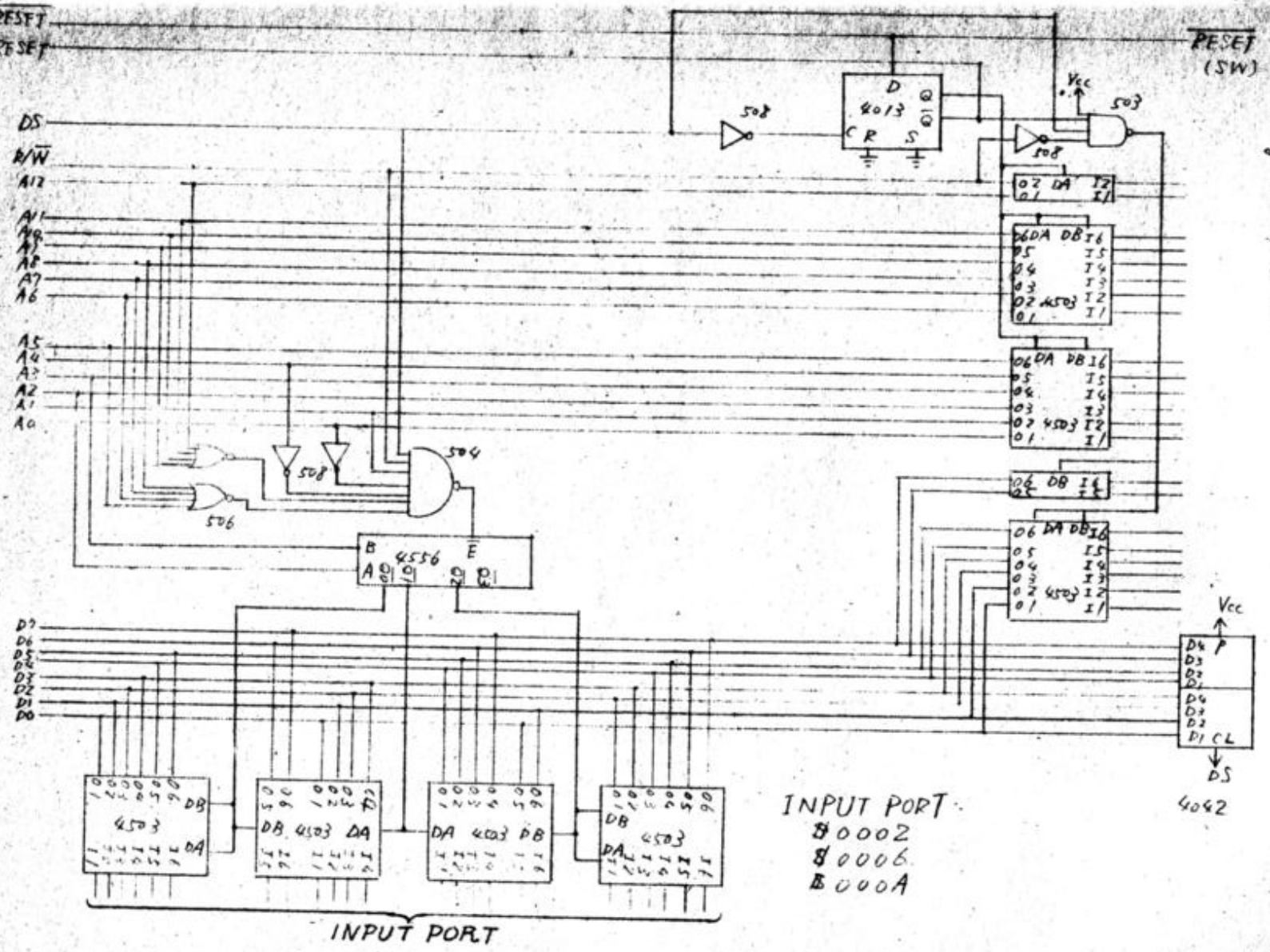

RESET
RESET
RESET (SW)
DS
R/W
A12
A11
A10
A9
A8
A7
A6
A5
A4
A3
A2
A1
A0
D7
D6
D5
D4
D3
D2
D1
D0
4013
4556
4503
4042
Vcc
503
504
506
508
INPUT PORT
INPUT PORT
0002
0006
000A

Fig 6 CPU-B

editor's free Area
巷に、あふれる。

16bit machine を 斬る。

▶はじめに L-kit 16 ありき。

L-kit 16と、言えば、日本マイコンの夜明けに、かのTK-80と、ワンボードの王座を割かち合った、マシンである。メモリ空間 64kw、16bit 並列処理。いかにも、すごそうに見えるし、1キーのPushで、アッセンブリが1step入力できたりワンボード System としては、良く考えてあった。

ところが、1word=16bit ということは、メモリを1word増設するのに、8bitマシンの倍の金がかかるということに他ならない。これは、当時[マイコン族=無銭家]という定義がまかり通っていたことからも、恐るべきパフォーマンスの悪さであった。また、Lkitの Processor(型名は、忘れた　)は、Byte Access ができない。つまり、Characterのアクセスができなかったので、文字を与うような Program では、非常な効率の低下をまねいていた。そのため、TEXT-Editorや、Tiny Basic は、どうしても、もたもたに、なっていた。そこで、私めは、しっかりと8万円握りしめて、TK-80を買いに、行ったのであった。1977年のことだった。TKは、今も裸だ。

当時、TMS-9900 とか、LSI-11 とか、いうのも、ちらほら表われていた。TMS-9900は、もたもたと、そっくりのアーキテクチュアをもった変態マシンだった。インタラプトやメモリアクセスのタイミングまで、全く同じようで、おかしな、気持ちがした。インストラクションのパフォーマンスも、もたもた程度だった。

LSI-11は、泣く子もだまる、DECのスーパーミニコン PDP-11の System 7(?.だったか)のプロセッサ・アーキテクチュアをすっかり焼いたものだった。もちろん PDP-11/70のSoftは全て動くので、自分で、PDPを作ることも、不可能では、なかった。LSI-8(PDP-8のLSI版)の製作が、月刊ASCIIに連載だったこともある。(日本データジェネラル(昔ミニコン)は、DECのまねして、Micro NOVAを作った。)

▶ 突然ですが 現在。

Zilogが Z-80を発表した。すぐ後、Intelは、8085と8086をアナウンスをして、両者とも、すごいChipのように言っていた。ところが、Intelで主力技術者が、みなZilogに逃げて、アナウンスどおりのSuper-Chipを作れるような奴は、全然いなかった。その証拠に、四苦八苦して、出した、8085は8080と全く変わらないアーキテクチャだったし、Chipが出荒したのも、Z-80が、日本のSharpによって超大量生産された、後だった。その後につづく、8086は言われていたような、Super Chipでは、なく、また、8086が、公表された すぐあとには、Zilog(AMD)の Z-8000が、本格的な、新世代アーキテクチュアで発表された。また、巷に、Chipが出だしたのは、両者、ほぼ同時期だったようだ。

その間 Westen Digitalは、Pascal Micro Enginを発表したりしていたが、実は、うわさでは、Zilogの新Chip(1977年暮)は、Z-80000と呼ばれ、Pascal(P-code)を直接実行できるということだった。また、Zilogも、新Chipの開発中に、技術者に逃げられ、Z-80000の開発が大幅に遅れたのは、知られるところだ。(Z-80000は、そのせいでポシャッタ?)

W.D社のMicro Enginは Micro Programなので、それほど目新しいことはない。DECも、実はがんばっていて、μCOM-16とか、μCOM-16000とかも、作っていた。μCOM-16000は、Micro Programで、Userが、自分でインストラクションを定義できた。μCOM-16000発表当時は、TTLをフラットパッケージにして使うのが、はやった!?

▶ 対決! Processor Architecture (やや Soft も割に見て…)

さて、僕は 8086 が、大嫌いなので、8086 基準にして、色々な、マシンをほめてやろうと、思う。

○ 8088: インテルが言うには、8bit で 8086 と同じ、パフォーマンスが得られるとのこと。同じなのなら、バスの配線が 1/2 で済む 88 の方が、絶体、有利なはずだ。16bit 得るのに、2度アクセスするので、実行するのが、86 の半分の速度になると思われるむきも多いだろうが、8086, 88 とも、高度のパイプライン処理を行なっているので、命令フェッチの オーバーヘッドは、フェッチユニットに吸収されてしまう。もし それを、おかしいと思うなら、86 が奇数番地へ、ばかり JUMP した時の速さを 88 と比べてみると、88 と 86 が、同じ パフォーマンスしか持たないことを理解できるだろう。

結局、他の部分(内部アーキテクチャ)は全く同じなので、極一部の、16bit メモリアクセス(それもバスインターフェイスユニットが、かなりのオーバーヘッドを吸収するが…)で、不利なだけで、DATA BUS が半分の 8088 の方が かなり有利に使えるはずだ。(プロの量産品でも、BUS が半分の方が有利!)

今、8086 で 自作マシンを作ってやろうと思っている人は、死んでも 8088 に、変えなさい。88 では、マルチプロセッサや、マルチタスク するような、大規模な System が 作れないと思うだろうが、それは、8086 でも同じなのだ!! ナント。この続きを読めば、それが解かるだろう。

○Z-80：なぜ、ここでZ-80か？それは、Z-80が、8080アッパーコンパチブルだからではない。始めに書いたとおり、Z-80は、8080を作った奴らが、作ったものだ。だから奴らの『8080の後つぎはこうしたい』という、意志が反映されているはずだから。

↑ずばり「ズィー・エイティ」と発音しよう。アメリカ人は「Z-80」と読む。

さて、アーキテクチュアだが、Z-80は、8086同程度のパフォーマンスしか無いと思う。それというのも、インテルが、Z-80を見て、それを非常に驚き、それに対抗するために、しゃかりきに作ったのが、86のような気がするからだ。確かに、86は、セグメントの概念を取り込んでいたり、大きなSystemも、小さなSystemも組めるように、配慮してあったり(ミニマムモード・マキシマムモード切り換え)するが、Instruction的には、Z-80と大差ないような気がする。16bit命令が強力なのは、当り前だし、太System Modeを選べる(Hard的にですね!)のも、Z-80より後でできたから、その程度は当然目新しいだろう。
また、セグメントも、Memory Managementを行なうO.Sか、ハードの支援無しでは、ただのアドレス拡張用のハードにしかすぎず、そんなものならZ-80でもすぐできる。現に、CromemcoやMP/Mは、朝飯前だ。

それより今までのSoftを全て使えるZ-80の方が絶対有利だ。アメリカでは(日本でも)Z-80のプログラムは、死ぬほどある。Zilogが、Z-80のインストラクションに乗除算をくっつけて、アドレッシング空間1MBにしたマシンにZ-800と名付けて出すといううわさを最近聞いた。うわさだけにしても、そんな話が出る程、Z-80のSoftが多いということの証しだと思う。

▶ ここで、突然、お断り申し上げますが私は実は(現在) MC-6809を載せた「なんとかマスター」を愛用しておるのですが、68系のことはさっぱり解からず、バスタイミングはおろか、「6809のマシン語恐い」と日々を暮らしているのです。「MB-6890 なんとかマスター」もその「Level-3 BASIC」が実は「TRS-80 (Model I) Level-2 BASIC」と正統な Upper Compatible だったので、買ってしまったのです。そのため、TV.Gameをするがため、今度は、「Game言語」のCompilerをわざわざ作って、68のマシン語(アセンブリィ言語も、マシンコードも含めた)から、逃がれたのでした。これが、なんのお断りかと申しますと、ぼくは、実は MC-6800 をを全然、知らんのです。アハハ…。やっぱり 8ビットは ええ。

しかし申し訳ない、

○6809:「なんで 8086と 6809を比べるんや」と、言われそうですが、それは、上に述べたとおりです。

だいたい、私は、68系が 安兼いなのだ、それというのも、68は、中が、俗にいう「しょうぎ倒しロジック」で、バタバタと動き、外界とは"0"と"1"とかいうclockで やって、つじつまを合わせているようだから。

しかしながら、Soft-Architecture は、8086など、及びつかない所もある。

1) レジスタが少ない。これは、8ビット系のあなたには、全くのまちがいのように感じるだろう。しかしながら、もともと おそい68は、レジスタのアクセスも、メモリの差もない。それでも、レジスタが少ないのは、石畳かて不利なようだ。ところが、いざ高級言語を動かそうとすると話は ちがってくる。例えば、BASIC インタープリタを動かす時、変数Area, BASIC-TEXT, STACKを ポインタするのに

最低3本のポインタレジスタが欲しい。それに反して、演算処理は RPN的に行われるので、Accは1本、または無しで良い。(Stackを演算に使うので、レジスタはいらない)。また、コンパイラ言語でも、ポインタは3~4本程度欲しいが、演算レジスタは…？もしあなたが8080のレジスタを全部使う様にオプティマイズするコンパイラを作れと、言われてそんな物が作れるだろうか。そこのところは、bit誌の少し前の奴にも載っている。(1981.9.P60). レジスタがたくさんあると、変にこった物を作ってしまうので、私の様な気の多い人間には、レジスタの少ない方が、割り切って考えられて、楽なのだ。そのうえに、スタックとのデータのやりとりが強化されていれば、RPN方式のStack Machineを形成すれば良いので、コンパイラが、ぐっとさばけるのである。(へたすると、式をtreeにして、演算順序をととのえて、それをおもむろに、OBJECTに落すという、大それた方法をとらされることもある)。「そんなんうそや」と思われる人も、いるでしょうが…実際に、インテルの最新MicroProcessor iAPX-432では、外から見ると(つまりUserから見て)レジスタの無い、完全なスタックマシンとなっているそうだ。これも、コンパイラ屋を苦しみから救う、有難い配慮なのだ。

本題にもどって6809の土俵合だが、なんとおあつらえ向きに、16bit Accは1本、ポインタは4本という、構成で、ポインタは全てArithmetic Stack Pointerに使えるのだ。(もちろん単なるStackでも良いが)そこで、16bitAccをStack Topにした、Stack Machineを形成した、OBJECTを展開するようにすると、かなり良質のコンパイラを作ることができる。

ii) メモリ・マネジメントについて。

6809には、segmentの概念は無いが、6829 MMUを組合せると、すごいものになる。しかし、6829+6809は、Z-8000とそっくりの動き(多分、Z-8000 Systemのコピーだろう)をするので、Z-8000でくわしく述べる。

Z-8000：はっきり言って、8086とZ-8000は、レベルものにならない。

その1）マルチプロセッサについて、

8086は、マルチプロセッサが、できるように始めから考慮されていたように、インテルは言っている。しかし、マルチプロセッサに限らず、プロセスが、同時に複数個が動く時、共有したリソースを、アクセスするには、その前に、セマフォをチェックして、セットしなければならないし、リソースのアクセスが終ったら、セマフォをリセットしなければならないのは、一般的に必要なはずである。そして、そのTest & Set, またReset, は非可分であるから、（セマフォの書き換え中に別なプロセスがチェックに来て、もしセマフォが、まだResetされていると誤解したら、1つのクリティカルに、2つのプロセスが入ってしまう。）マルチプロセッサの場合は、少なくとも、同一のセマフォを2つのプロセッサが同時に、アクセスすることが無いようにしなければならない。

その点で8086はプロセッサ間の同期をとる信号線が全く無いので、HOLDを使うしかないが、Z-8000はMI, MO (Mulei Micro In/out) でプロセッサ間の同期がとれる。すばら!!

その2）マルチタスクについて、

インテルは、8086は、マルチタスクも考えてあると言っているが、なぜ、System Mode（O.S用）とNormal Mode（一般的User用）の、区別も、ないもので、マルチタスクできるのか？ 8086は、全くの外付けHard無しでマルチタスクをしても、一つのタスクが暴走すると全Userのタスクが、Hangして、ついには、O.Sをも、右往左往する可能性がある。なにしろ、全メモリ空間が、まったく同じ条件で放っておかれるのだから。

もし、本気でマルチタスクをやりたいなら、外部で、1度切り換えると、次には、ResetかInterrupもでしか、もとに、もどらないメモリバンクと、適当な条件でアクセスすると、ResetかInterrupもを発生するI/O Poreもをつけてやって、先のバンクに、O.Sを入れるしか、ないだろう。そして、System callは、I/O Poreアクセスで行なう方が良い。これなら、ユーザーのタスクが暴走してもO.Sのメモリバンクは荒らされないし、I/O Poreをアクセスしてしまっても、その時はO.Sに帰るので、うまくいく。　ところが、Z-8000では、始めから、SystemとNormalの2Modeがあるし、Normalでは使えない特権命令も用意されているので、真に、マルチタスクが可能である。

実は、6809+6829でも、上の方法で、Systemモードと、Normalモードを切り換えられる。この2chipは、本当に、Z-8000を、そのまま縮小したような、パフォーマンスを持っている。

そのう）メモリマネジメントと仮想記憶

世間で言われている通り、セグメンテーションを用いることの利点は、モジュールのリロケータビリティと、それにともなうダイナミック・アロケーション性であるが、それによって、VOS（仮想記憶を用いるO.S）が、非常に、活動しやすい。しかし、8086のは、それだけのものの為のに比べ、Z-8000+Z-8000MMUでは、セグメントの大きさが自由に変えられる。セグメントの物理アドレスを自由に設定できる。セグメントにアトリビュートを与えられる。そして、セグメントのレンジがオーバーしたり、アトリビュートがミスマッチした時、MMUがチェックしていて、TRAPが起こる。つまり、セグメントが切り換わるたびに、VOSが間に立ってチェックをしたり、レンジのオーバーに気をくばっていなくても、MMUのHardが自動的に知らせてくれるので、VOSの負担が減り、また、メモリアクセス時のVOSによるオーバーヘッドも少なくなる。

なんと、Z-8000の強力なことか。それでも、8086には8087なる浮動用Co-Processorがあると言われるだろうが、Z-8000とも、FPUなるものが、アナウンスされている。

それにしても、6809 Systemのすごさは、Z-8000の縮小版から、たったの3ヶ月で、Color Computerになっただけで、ますます、そのおもしろ味を増してゆくようだ。

しかし、やはり、ミニコン並のSystemを作りたいなら、Z-8000程度のものでないと、Hard的にも、Soft的にもしんどいし、できても、使いものにならないだろう。68000にも、MMUがあるらしいが、モトローラは、口だけなので、供給はまだまだ先のことだろう。

以上、えんかげんな事を述べて来たが、最後にやはり一番すごいのは、Sweet-16なる16bit Processerだろう。これは、世界的に名をはせた、Apple II (INTEGER Type)に登載されている、Super Processerである。ぼくは、このSweet-16とAPPLE-INTEGER BASICの組み合わせ以上に、Betterなものを思いつかない。(「MINI Assemblerの方が、弓強力だ」との声あり。)

↖なんのこっちゃ。

NOV. 17 '81

(Editor S. Takeoka)

今では"前"部長となっています...

表題は編集で つけました.

私は 四回生であり、本来ならば このような原稿などは 書かせてもらえないのですが、編集長から、出稿数が足りずに困っている という話を聞きましたので 厚かましくも ここに 掲載して頂くことになりました。最近では、部室でも年寄り扱いされ、誰も遊んでくれないので腐っていたところなので、この原稿によって頼りにされているという実感を、久々に満喫してみたいと思います。

一応、私も部員数のうちであり、今学園祭の展示で使用するマイコンの設計をさせてもらいました。CPUの機種については、"あの有名なCOSMACを一度使ってみたい"との声が高かかったのですが、私は モトローラのMC6805を推しました。COSMACを主張した者の推薦理由としては、

1. 価格が安い.
2. 入手が容易で、大阪でも買える.
3. 周辺のファミリが揃っている.

その他、どれをとっても もっともなものでした。それに対し、6805の側は、資料もなく、唯一の推薦文句は、今までどんな本にも載ったことがないので、見栄が張れ

る。ぐらいのものでした。しかしながら、結局6805でできあがっているのが、四回生の権威でしょうか。いずれにせよ、このことにより、私は、金は出さずに口だけ出す四回生、とのこれまでの評価をさらに高めたようです。さらに、マイコンや、その他の設計については色々とお手伝いさせて頂きましたが、どちらかと言えば、後ろからやれやれと掛声ばかりかけて、自分は陰っこで寝そべっていました。

学園祭までの期日が切迫してくると焦り出すのは例年同じですが、今年は優秀な創作スタッフが各々頑張って展示品目を創ってくれているようなので、今から楽しみにしております。毎年、ソフトウェアは「四回生頼み」となるのが常で、今年も一部ぼつぼつそうなっていますが、外目には嫌でたまらない、という風を装いつつ、内心の「相手にしてもらえて嬉しい」気持ちを精一杯抑えて、思いきり恩に着せてやろうと考えています。

森本 英一（電子Ⅳ）

☆ ☆ ☆ ☆ ☆ ☆ ☆

MC146805を使って

ワンボードマイコンの製作

今回、機器制御に使用したワンチップマイクロコンピュータMC146805、及び実際に設計、制作したワンボードマイコンについて若干の解説をさせて頂きます。

MC146805は、モトローラ社で各種出されているMC6805シリーズの一品種でMPU、RAM、I/Oポート16ビット、及びインターバルタイマがコンポーネントされたワンチップマイコンです。命令はMC6805シリーズに共通なもので、その特徴を一言で申し上げますと、非常に直交な覚え易い命令体系、と言えましょう。通常の使用目的では明らかに必要ないと思われるような、インデックスレジスタを操作する命令のインデックスアドレッシングモードや、使い道に困るようなジャンプ命令のバリエーション等も、命令の直交性を維持するために、機械語のしかるべきビットパターンのところにちゃんと存在しています。また、スタックはサブルーチンの戻りアドレスの保持と、割り込み処理のみに用いられ、プログラムで自由にレジスタをプッシュ・プルすることはできませんが

これも、ある意味では、命令を判り易くするのに貢献している、とも言えます。更に、非常に強力なゼロページのビット操作、ビット判定命令などとも考え合わせると、比較的小規模な機械制御目的に対しては、まさに絶好のMPUであることを、モトローラ社の回し者として宣伝しておきます。

さて、このMC146805は、C-MOSであり、周辺ICも全てC-MOSを使用すれば、電池で稼動させることができます。でき上ったマイコンの消費電力を測定してみますと、プログラム実行時で約15mA、パワーダウンモード時で約300μAと一応満足のゆく値が得られました。パワーダウンモードにしておきますと、少くとも1ヶ月程度は電池を交換せずにすむものと考えられます。

このワンチップマイコンは、組み込み用として設計されていると考えられ、当然DMAのサポートはありません。C-MOSのROMが一般に入手困難な現状では、メモリの読み出し、書き換えにハードウェアパネルを使用せざるを得ないので、これには頭を悩ませました。結局、マイクロコンピュータの外にスリーステー

トバッファを入れ，RESET及びデータストローブと同期させるという極どい方法で解決しています。

現在のように、完成品の高性能パーソナル・コンピュータが安価に入手可能な現状では、このようにマイコンを自作する機会は非常に少ないと思われますので今回の場合は、どうしても電池で稼動させたかったので"やむを得ず"制作したのですが、できあがってみると、完成品のマイコンにはない愛着がわき、フロッピーディスクでもとりつけてやろうか、などと考えたりもしています。

目下のところ極めて快調に動作しており、誤動作等全く見られないことを申し添えて、まだ名前のない私達の新しい仲間の御紹介を終ります。

(森本)

♡♡♡♡♡♡♡

Personal Computerの
データ通信・ネットワーキング
—その意義を考える——— Take.

パーコンでの遊び方(使い方)も色々あるが、その1つにデータ通信用の通信器とするのがあると思う。もちろん、それに反対する人もあって、「ディスクとプリンタ、それだけあれば完結した系なので、外からデータをもらったりする必要はない。」と言い切る人もある。また、その裏付けとして、「日本では、T.S.S.が流行ってない。」と言われる。実際、TSS端末は、一般の会社のOfficeや、高校で見られたりしない。しかし本当にそうだろうか？

日常生活でもデータ通信ができると便利なことが多い。

実例 その1) ある日、実験のレポートを書いている。どうも結果が変なので友人Aのところへ電話する。「おい。あの実験の結果、どうやった？」 すると、友人Aは既にデータ整理が終って、結果がディスケットの上に落してあった。しかし、通信機能が無いので口頭でやり方を説明してもらうしかなかった。

その2) 友人Aは、自分用に文献探索システムも作ってあった。俺は、ハードの設計をしているとき、昔のインターフェイスに似た記事のあったような気がした。友人Aに電話して、そんな本を探させた……通信など出来ないから、もちろんAにマニュアルで操作させた。すると、友人Bの家にあることがわかった。友人Bに電話すると、その本の内容は、ワードプロセッサでディスクの上に落としてしまって、その本は捨ててしまった。と言った。ディスクか、プリ

ントアウトをやるから取りに来いだと。こんな時に通信が出来たらなあ。

その3) 自作マニアの友人Cは、ビンボー人で、メモリも、ディスクもプリンターも持っていない。そこでいつもコンパイラや、アッセンブラを借してやるが マシンコードのダンプを手で打ち込んでいる。しかも、自宅から我々の家まで距離が遠い。こんな時、電話一本でつなげれば。

上の様な時、ものぐさな私めは、家から出ずに、他の家と、DATAをやりとりしたいと思う。ディスクというものを使えば、確かに大量のDATAを安上りに交換出来るので非常に有効だと思うが、ディスケット一枚分もないデータ量、つまり、データの通信時間がガマンできるぐらい短かければ、(金銭的にも) 直接に会わないで済む電話か、そのような物の方が楽だ。

しかし、上の例のような通信をするときには色々な問題がある。まあ、ハードはシリアルインターフェイスを使えばそんなに苦にならないが、一番の問題は、肝心のデータのフォーマットである。例えば、PC-8001とMB6890は、テープインターフェイスのハードはほとんど同じで、1word(1Byte)ごとの、フォーマットもコンパチビリティがあるのに、その上位のフォーマット、Blockなり Headerなりの形式が全く違うので、PCの作ったテープをMBが読むことは出来ないのである。

そこで、Protcol というものが登場する。プロトコルというのは、プログラムならどう、データならどう、さらに数値ならどうで文字ならどう、また、それらの識別子がどう、ブロックがどう、また1wordやりとりするタイミング、電圧、電流、1Blockごとるタイミング、またその上位のパケットがどうのとか、通信に関するありとあらゆることを決めた規約で、時には、それらを司さどる基本的な手続き(プログラム)をそう呼ぶこともある。もっとも、最近のプロトコルは、ハードは把先にある規格を用いるので、あまり下位のことは気にしないしかしこのプロトコルが非常に重要なのはPCとMBのテープの例を引かなく

くても解る

しかし、もっと重要な問題点は、本当に、共有する価(相互に通信する人)値のあるデータがあるかどうかだ。実は先に出した3例は、頭を使わないで出したものなのだ。

折りしも、アメリカではXerox, Intel, TRS(だったか?)が共同で、データ・サービスを始めるはずで、少くともXeroxはSmalltalk likeなマシンを、またTRSはColor Computenをそのターミナルとしてもってくるはずだ。つまり、どこかが、ON-LINE通信すれば得になるその情報を公開してくれるのだ。

それでは、日本でパーコンを使っている奴らは、通信しても意味がないのか? それなら、自分達で共有して有効なものを作れば良いのだ。それには、色々なデータもこることながら、やはり、コンピューター・マニアとしては、プログラムを共有したい。しかも例3のようにオブジェクトを生成する場合、クロスコンパイルも良いが、どうせなら、オブジェクトも共有したい。そこで、突然UCSD-Pascalのような言語が注目されるのだ。はっきり言って、今日本のパーコンで、テープなりディスクなりのメディアが異機種間で共有出来るものは全くない。(もっともPC-8001, MZ-80K/CとMB6890は、ハード的なトラックセクターの構成は同じようなので、無理にたら読み書きできる。)したがって、CP/M, UCSD-Pascalなどが動いても大したうまみはないのだ。(それでも、販売元が一生懸命めい、それ用の(CP/Mならば/M用の)既存Softもそのマシンで動める形にして売り出してくれるから多少はましだ。)だからPCのUSERがUCSD-Pascalを動かしても、APPLEのUSERのUCSD-Pascalとはプログラムを共有するすべがない。そこで、やっぱり通信をして、プログラムでもDATAでも共有するしかない。それに、そうでないともったいない。そして当クラブでは、そういう壮大な計画のもとに、マクロなプロセッサを核とする構造化高級言語を開発中なのです。

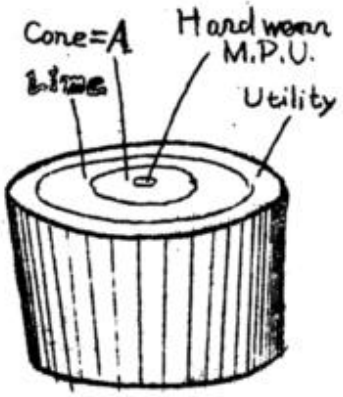
異種のMicroProcessorでOBJECTまで共有できる.

Lime®

Computer Language

Limited Expression

Limeは、ALGOL-60の血を引く古いTypeの構造化言語です。しかし、そのためコンパクトでポータビリティが良くなっているのが特徴です。



SYSTEM概念図

Lime自体は、コンパイラ言語ですが、Lime Systemの核には、常にprossor Aがあり、全てのSystemはA上で稼動します。従ってLimeの生成するOBJECT codeもProssor Aのマシンコードとなっています。これは複数個のProssorでOBJECT codeを共有する場合、どのProssorでも例外なく実行が行えることを示し、また実際にコンパイラを起動できるものは最低一台であれば良いことを示しています。上記のことは、これから益にパーコンが増え ON-LINE dataは通信が頻繁になっている現在 非常に有効なことと思われます。

リリース予定；本年末頃
価格(予)；ライセンス料 100万円
発売元(予)；電子科 TV Game 同好会
製作スタッフ；奥田さん 澤谷クン 竹田
田中MASAサマ、淵原チャン
(冗談も入っています.)

"エディターに「あんた部長やからなんか書け。」と言われたから、しょうがなしに思いついた事をならべたエッセイのつもりやけど、ほんまきやろか"

よしだ あつし

たぶん高校2年の冬であったと思う。私がシングルボードコンピュータなる物を見たのは。日本橋の共立のショウインドの前にすわりこみ、TK80をじっと見つめていた。この時私はまだコンピュータを"魔法の箱"と思っていた、だからどうしてこんなに小さくなったのだろう。これが有ると何んでも出来るんだろうな、と思いながらくいいる様に見つめていた。この時以来「おれもコンピュータを持ちたい」と言う淡い恋心に似た物を持ったのである。

以来もう5年を経過しようとしているのであるが、この5年でマイコンは大きく変ってしまった。LSIがむき出しの板切であったものが今やきれいなボディにおさまっている。RAMだってそうだ。8Kあれば神様であった。それがMZなどは60Kなのである。機能に対

するコストは目に見えて下った。そしてハード的には５年前のミニコン、いやある面では大型の機能以上の物を持ったマイコンを、私たちは少々高額であるが手にする事が出来る様になったのである。
そしてその事を大げさに（私はそう思う）マスコミは報じ、まるでマイコン機器が攻めて来る様に言う。しかし本当にそうなのであろうか。マイコンがわからなければ社会生活が出来なくなるとでも言うのであろうか。今考えてほしいのはマイコンはだれのために作られるのかを考えてもらいたいのである。やはりそれは私たちのためであるはずだ。私たちを非人間的労働から開放するためにではないか。私たちのためであるのなら、それは私たちに苦痛を感じさせてはならないはずである。使いにくいとか、わからないとか、そう言うイメージを抱かせてはならないはずである。だからOA（オフィスオートメーション）だって、将来家庭に入るであろうホームコンピュータだって”コンピュータ＝ムズカシイ”と言う

イメージを感じさせずに、相談をしたら答えてくれるテレビとでも言う、身近なコンピュータを意識しない形にならなければならないはずだ。たとえば電卓だって立派な計算機（コンピュータ）だ。だがこれをコンピュータだと思って使っている人がはたして何人いるであろうか。（Editor注 カリキュレータ＝電卓は、コンピュータではありません）

しかし日本のメーカーはこのコンピュータをコンピュータらしくなくする方法に、あまり目を向けていない様に思う。日本は現在多分世界で最も高い機能を持ったマイコンを作っている国であろう。しかしこれを使うには、プログラム（ソフト）が必要なのは当り前である。だがこれらのメーカーは、どれだけのソフトをユーザーにもたらしただろうか。高い機能を持てばそれだけソフトを書く事は煩雑となるし、高度な機能のソフトを要求、期待されるであろう。ちゃんとしたソフトさえ有ればコンピュータの中身など知らなくとも、ただキーボードなりをタタケバ何かをしてくれる、便利な箱である。だがソフトが無ければ全くの無能なた

だの箱、いおやっかい物と化すであろう。素人にソフトを書けと言ったって無茶である。現在特別なソフトを書いてもらうには目の玉の飛び出る様なお金を取られてしまう。今はまだユーザーになろうとする人たちは無知に近い、だからそんな事も知らずにハードだけ入れればすぐにでも使えると思っている。しかしもう少しカッコクなればソフトが大きく選択のウエイトを占るであろう。その目のためにもメーカーは大型機にはらっているのと同じ様にマイコンでもソフトに力を入れてほしいのである。さもなければメーカーはその時痛い目を見るであろうし、マイコンの未来も真暗である。

ところでこの5年目ざましく発展して来たハードは今後どうなるであろうか。あと1年〜1年半すれば16ビットのマイコンが出ると私は思う。だが16ビットマシンとしての機能は十分に発揮出来ないであろう。16ビットマシンがその機能を十分に果すには、大きなRAMとバブルやディスクの様な外部メモリーを持ったシステムとなってこそ

であろう。(そしてもちろんそれらを管理するOSも) だがこの様なシステムを一度に買うには高額すぎて、パーソナルとなりにくいと言う問題が生じるわけである。だからホビストが使うと言うよりは、出来上ったシステムに使われるであろう。オフィスにおいては今ミニコンがやっている様な事を、家庭においては情報システムのターミナルや空調、防犯などの家の管理を一手に行なう様なシステムとして活躍するだろう。ところで8ビットのパーソコンは今が最盛であろう。これ以上システムアップするのなら16ビットに成って行くだろう。だから8ビットは16ビットが出て来ると、もっと親しみやすいコンピュータとなって行くだろう。前にも言ったコンピュータを意識しないコンピュータへの方向にである。その点ではPC6000はその先がけとして表価したい。あれはみるからにオモチャである。しかしコンピュータの一般家庭進出と言う点で先駆者として大きく貢献するであろうし、小学生などのこれからの子供たちにコンピュータに対

するイメージを身近にするであろう。そうだその点ではインベーダーゲームなどのテレビゲームも表価されるべきであろう。8ビットは技術も確立して来ている。だから安価に供給出来るであろうから、一般家庭進出の先兵となりうる能力と必然性を持っているわけである。

CPUはそれらの持つ能力に見合った方向へと進んで行くだろう。それはあたかもダーウィンの進化論に従う様に。大自然が空気と水と土からわれわれを人間にまでして来た様に。彼らも人間によって空気と水と土から創造され、進化して行くであろう。

<悪のりその1　読者のみなさんへ>

このエッセイは独断と偏見で書いている。だからここに書いてある事を人に知られたり、もしくは他人にしゃべる事は非常に危険である。だが君たちが危険にさらされても一切当局は感知しないのでそのつもりで。なを このエッセイは自動的に消滅をする ことはない。よって自分で食べてもらいたい。

でわ読者の幸運を祈る。

↑Editorはこれを「は」に変えたかったが、ギャグだと理解した。

＜忍のりメのス　特別付録　筆者の自己紹介＞

名前　吉田　篤（よしだ あつし）

工業化学科の3回生である。容姿端麗眉目秀麗では無いのだ。身長170cm体重は㊙なのだ、少々太めである。何故か知らないが部長をやっているのだ。たぶん我がクラブには まともな人間が私しかいないからだろう。だからいつも私は苦労をしている。趣味は旅行、今北海道に取り付かれている。2回も行ったのにまだまた行きたいのだ。弟は私と旅に出ると「修行をしてるみたいや。」と言う。それは旅に出ると私がほとんど飲まず食わず、車中泊、1日中列車の中のパターンが多い。要するに電車が好きなのである。だから電車の写真もよく撮る。しかし一度女の子も写してみたいな。メッチャンコかわいい子を。マイコンは道楽（アソビ）である。だからマイコンは持っていないのだ。あれはレポートを書くのの毒じゃ～。（一部に嫉妬が有った事をお

注「憑く」が正しい

おびします。)
ジャジャジャジャン! 女性読者のみなさん、ファンレターの送り先を書くのだ。京都市左京区修学院室町9-1中徳荘 TEL 075-721-6648なのだ。ヤロウのファンレターはいらないのだ、かわいい女の子オンリーなのだ。なんちゃって変な事ばっかり書いておかしんやなりやろか? 実は後半の入ードうんぬんからほんの少し酒を飲んで書いているのだ。私がこんな事を書くなんてやっぱり酔ってるのかな。しかしもう3時間前にビンク一杯しか飲んでないのになあ。しかしおかしい。なんぼ考えてもわからへんから早よ寝よ。もう午前3時をまわってしまった。明日は実験だ、居眠り出けへん、眠むたいやろなあ。また変なこと書いてる。ああもう寝よ。
自己紹介がめちゃくちゃになってもた。ごめんなさい。

編集後記

▶ 当クラブ、発足当時のことは、話しでしか、知らないが、[illegible]は、部員数も増え、財政的にもRichなので、大きな[illegible]できた。その成果と、また個人の日ごろの[illegible]を、外界へ知らしめたいであろうと思い、この小冊子[illegible]した。しかし、皆、パープリンなのか、外界との接触を[illegible]のか、原稿は、ほとんど集まらなかった。外界とのコミュニケーションをしたがらないのは、勝手だが、いくら自分の中だけで思ってみても外から見れば、変人と同じだし、他人へ、利益を与えることもない。日頃Programと、いう割で情報を表記している人間が、日本語で、自分の考えを表記できないのは、問題だと思う。

▶ 話しは、かわって、巻頭に部長も書いているが、マイクロコンピュータの性格上、そのニーズは非常に多様化している。そして、その多様化の中で、当クラブの構成員も、その目的は多岐に渡っている。昔なら「ハードを作る」という目標が、第1ステップとしてあったものが、今は、「使う」ということで、始まるのだから、当然とも言える。今日、一応「ハード」、「System Soft」、「人工知能?」という形で、欲求を満たしてはいるものの それらの統一目標がないため学祭に対しての、乗りも、もう一つなのだ。いわば、このモラトリアム・クラブをどんな方向のコンピュータ・クラブにするかは、現在の構成員に任されているのだと思う。せめて「TVゲームクラブ」には、したくないものだ。

▶ 記事中に、茶谷さんが人工知能について、書かれている。私めも、いつもは「Systemいじり」をしているけれど、実は人工知能に興味を持っているのだ。人工知能で使われる言葉は、書かれた本によって若干の違いがあるので それに関する注や、私めの見解を書いた記事も載せたかったが、時間の都合で無理だった。人工知能をやるなら「Lisp」を、勉強さい。

Editor
竹岡尚三

NOV. 18, 1981.

Sweet-16 は
何処へ・・・・

-Take

CTTU

Thank & Love