心理学実験におけるマイクロ・コンピューターの役割について ・実験制御装置としての利用法の実際とその問題点

<table>
<thead>
<tr>
<th>著者</th>
<th>藤井 正也</th>
</tr>
</thead>
<tbody>
<tr>
<td>雑誌名</td>
<td>人文論究</td>
</tr>
<tr>
<td>巻</td>
<td>1</td>
</tr>
<tr>
<td>号</td>
<td>1</td>
</tr>
<tr>
<td>ページ</td>
<td>36-48</td>
</tr>
<tr>
<td>発行年</td>
<td>1986年10月25日</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/10236/5888">http://hdl.handle.net/10236/5888</a></td>
</tr>
</tbody>
</table>
心理学実験におけるマイクロ・コンピューターの役割について
—実験制御装置としての利用法の実際とその問題点—

藤井正也

十年一昔というが、この十年の間に関連領域における急速な研究の展開とともに、実験の制御装置の進歩にも目を見張るべきものがある。我々の研究室は、心理学的事象について発せられる経験的疑問（empirical question）に対して、実験によって解答を与えていくことをその特徴としているが、1970年代半ばから現在までの間に、実験の制御装置がトランジスターを中心とした回路からIC（集積回路）を中心としたものへと代わり、さらに最近ではマイクロ・コンピューター（以下、マイコンと略記）が盛んに用いられるようになってきた。

マイコンは条件さえ整えば、スピード、精度、耐ノイズ性等々、制御装置としてすぐれた特徴を有しているが、十分にそれを使いこなそうとする時、その使用者に心理学者ではなくて哲学者（技官）になることを要求する。マイコンを動かすためのソフトウェアの開発等々に非常に大きな労力と長い時間を必要とされることがあり、下手をするとマイコンと格闘するのをやめに陥ることとすらあるのである。

さらに、せっかく苦労して作成したソフトウェアも、その管理が大きな問題となる。自分自身が書いたプログラムでさえ時間がたてば何をしているプログラムなのかがわからなくなる（美濃, 1980)。ましてや他の人が作成したソフトウェアで、ドキュメンテーション等がきちんと保存されていないようなものなら、作成者以外の人間にとってはそのソフトウェアは無意味づのかたまりで
心理学実験におけるマイクロ・コンピューターの役割について

しかしながら、また、制御のために書かれたソフトウェアは、インターフェイス回路を含めたハードウェア全体と深く関わっており、この関係の資料が手元にないとき、やはりそれは無意味なものとなってしまう。

このように、マイコンを実験制御装置として用いる場合には多くの問題が挙げられるが、小論では、筆者がこれまでに関係してきた動物実験のためのシステムを例にとって、心理学実験における制御装置としてのマイコンの利用の実際を記述し、その後にシステム開発の際の問題点に触れ、最後に現在考えている理想の実験制御システムについて述べることとする。

I これまでに関わってきた制御システムの概要

Table 1 は、今までに筆者が関わってきた実験制御システムの一覧表である。表からも明らかのように、現在までに4種類のマイコンを使用してきたわけであるが、これらに共通する3つの特徴を挙げることができる。第1に、いずれのマイコンも80系の8ビットCPUを中心としたシステムである。第2に、どのシステムも一台のマイコンで複数のboothを管理している。制御対象が複数であるのは、実験の性質上、平均値あるいは中央値を代表値とするグ

<table>
<thead>
<tr>
<th>制御対象</th>
<th>実験の内容</th>
<th>システムの開発期間</th>
</tr>
</thead>
<tbody>
<tr>
<td>TK-85 (NEC)</td>
<td>パー押し反応事態での条件性抑制</td>
<td>1981年4月-8月</td>
</tr>
<tr>
<td>SORD M100 ACE IV (SORD)</td>
<td>DRLスケジュールがラットの行動に及ぼす効果</td>
<td>1982年4月-7月</td>
</tr>
<tr>
<td>KGbox-24 2台</td>
<td>24時間連続実験下でのラットの行動観察</td>
<td>1981年10月-1982年3月</td>
</tr>
<tr>
<td>PC 8801 (NEC)</td>
<td>パー押し反応事態における条件性抑制</td>
<td>1982年12月-1983年5月</td>
</tr>
<tr>
<td>Staff† (B・U・G)</td>
<td>同上</td>
<td>1985年12月-現在 (主として移殖)</td>
</tr>
</tbody>
</table>

†Staffは、札幌にあるB・U・G社製の工業用マイコンである。詳しくは後述の10ページ以下を参照のこと。
ループ・データを収集しなければならず、一度に複数の被験体をrunさせて実験の能率を高める必要があるからである。第3の共通点は、これこそまさに第2のそれから派生してくる最大の問題であるが、BASIC等のいわゆる高級言語ではなくて、機械語（machine language）でプログラムを書かなければならないことである。BASICでXとYを足し合わせようとすれば、単純に、
Z = X + Yとして演算結果をCRTかプリンターに打ち出せば終わりである。
しかしながら、機械語で同じことをしようとすれば、Fig.1のように門外漢には一目見ただけでは何をしているのかがよくわからないようなプログラムを書かなければならない。

LD A, X ; Acc ← X
LD B, Y ; B ← Y
ADD B ; Acc ← Acc + B

Fig. 1 X + Yを実行するための機械語プログラム例（1バイト精度）。
ニューモニックは、Z-80用のサイログ形式に準拠している。
図中のAccはAccumulator、Bはregister Bを示している。

なぜ、機械語なのであろうか？理由は簡単で、プログラムの一一行を逐時解釈して実行していくBASICのようなインタープリター型の言語では、実験のスピードにマイコンの処理速度がついてこれず、反応の読み落し等々が生じてくるからである。特に、複数のboothを対象として十分なスピードと精度で反応の計測や刺激提示を行いながらとする時、BASICでは全くお手上げで、機械のレベルまで人間が下りていってマイコンが実行するプログラムの1ステップ、1ステップを人間が手で書かなければならない（たとえば、藤、1982、pp.520-521）。しかもハードウェアを含めたシステム全体を考える時、ほとんどの絶望的とも思えるほどの肉体的、精神的負担がシステム開発者にのしかかってくるのである。

以下に、紙数の関係から、PC8801を用いたスキャナー箱での条件性抑制実験に限って、インターフェイス回路を中心とするハードウェアとソフトウェアとの接点であるバー押し反応の検出を取り上げ、筆者のストレスに満ちた体験の
一部を読者の方々にも経験していただこう。

II 制御装置としてのマイコンの利用の実際

スキナー箱での条件性抑制実験は、2つの部分から成り立っている。1つは、一定の強化スケジュールのもとで食餌（通常は45mgの乾燥固型飼料）を得るために、ラットが行うバー押し反応を記録する部分である。今1つは、このようなバー押し反応がなされている際中に提示される、光や音などの条件刺激（CS）あるいは電撃などの無条件刺激（US）あるいはその両者によって引き起こされるバー押し反応の抑制の記録である。

Fig. 2 は、バー押し反応を検出するための自作のインターフェイス回路を示している。図にも明らかのように、スキナー箱内のバーがラットの何らかの反応によって一定の圧以上で押し下げられると、これと連動しているマイクロ・スイッチが閉じ、monostable multivibrators (TI 74121; 以下, one-shot と略記) が作動して約15msecの間整形されたパルスがマイコンのProgramable
Peripheral Interface（以下、PPI と略記；型式 μPD 8255 AC-5）の人力ポートのうちのある特定のビットに与えられる。

一台のマイコンで複数の booth の反応を検出するために、筆者のシステムでは 10msec 間隔で発生する割込み毎に PPI の人力ポートのデータをチェックしている。4 台のスキャナー箱を対象として実験を行っている途中で、booth 0 でパー押し反応が生起した場合を例にとって、その考え方を示そう。

Fig. 3 の左側を見ていただきたい。booth 0 の反応はインターフェイス回路を介して PPI の最下位ビット（ビット 0）に与えられており、そのデータは回路上の定義により 1 である。一方、ビット 1 から 3 までは booth 1 から 3 までに対応しており、そのデータはいずれも 0 である。

Fig. 3 パー押し反応を検出するためのソフトウェア。
（左側のパネル：booth 0 で反応が生じた時の模式図
右側のパネル：Acc のソフトによる反応の有無のチェック過程
■は、RRA 以前の CY のデータ）

ソフトウェア側では、割込みがかかること、まず IN 命令により PPI の人力ポートの状態を CPU のレジスターの 1 つである Accumulator（以下、Acc と略記）にロードする。かかる後に、RRA 命令によって Acc にロードされたデータを 1 ビットずつ右へシフトして、キャリー・フラグ（CY）がセットさ
心理実験におけるマイクロコンピューターの役割について

Fig. 3 の右側は、このチェックの過程を模式的に示したものである。1 回目の RRA 命令で、booth 0 に対応しているビット 0 のデータが 1 であることにより CY がセットされる。CY がセットされる時、当該のビットに対応する booth で反応が生起しているとソフトウェア上で定義すれば、まさに上の方は反応が生起していることを示しており、後続のルーチンで反応数を数えているカウンターを +1 する等々の必要処理がなされる。一方、2 回目以後のシフトでは CY はいずれもセットされず、定義により booth 1 から 3 まででは反応は生じていないとして、それぞれ次のルーチンへ移っていくのである。

ところが、これだけでは問題が生じてくる。すなわち、筆者のシステムでは PPI に印加されるパルスの幅はタイマーの更新時に用いられる割込みの時間間隔よりも長くなるように設定されている。パルスの幅が 80 msec で割込みが 10 msec 毎に発生する場合を考えてもよい。もし上のように、PPI のデータを Acc にロードして RRA 命令により CY がセットされるか否かを割込み毎に

![Diagram](image)

Fig. 4 Chattering Buffer (CB) による反応検出ルーチンのコントロール。反応検出と同時に CB のデータを 1 にして、以後割込みが 8 回生起する間、反応の検出を行わないことを示している。
毎回チェックすれば、コンピューターは1回の反応で生じる80 msecのパルスによって8回の反応が生起したと勘定してしまうことになる（Fig. 4・上部）。これは不合理である。

対策としてはいくつかの方法が考えられるが、筆者は80 msecのパルスのうちの最初の10 msec足らずの部分のみを1反応として検出し、残り70 msec余りの部分をソフトウェア的に無効にする方法を採用している。すなわち、Fig. 4の下部に示されているように、各booth（各ビット）に対応して1バイト精度でChattering Buffer（以下、CBと略記）を予約しておく、このBuffer内のデータによって反応の検出の有無をコントロールするのである。具体的には、割込み時に反応検出のためのルーチンの先頭部分でCBのチェックを行い、CBが0であればPPIの入力ポートのデータをAccにロードする。そしてRRA命令によってCYがセットされれば、反応が生じると判断し、以後の一定時間反応の検出を行わないようにするために、CBの値がある特定の値に書き替えられる。次回の割込み以後、以前と同様、まずCBの値がチェックされるわけではないが、CB≠0であるのでPPIの入力ポートのデータはAccにロードされず、CBの値を-1しただけで次のルーチンへ移っていく。そして再びCB=0となった時に、入力ポートのデータのチェックを再開するのである。したがって、たとえば、反応の取り込みに伴ってCBの値が8に書き代わるのなら、以後の80 msec（割込み1回分10 msec × 8）はPPIに印加されるパルスがたとえ1であったとしてもそれは無視される。

III システム開発の際の問題点

上記の例のように、マイコンで実験制御を行うためにはハードウェアとソフトウェアとの接点でプログラムを組まなければならないが、以下に、スキャナー箱のシステムに限って問題点を整理してみよう。いくつかのことが考えられるが、(1)インターフェイス回路、(2)ソフトウェアの開発環境、そして(3)マイコンそのもののハードウェア、という3点がシステム開発の際の問題点として浮か
び上がってくる。

まず第1のインターフェイス回路であるが、これはさほど大きな問題とは思われない。なぜなら、TTL ICに関する少しばかりの知識と経験があるなら、Fig.2の回路図が何を意味しているかは比較的容易に理解できるからである。

より重要な問題は、ソフトウェア開発のための環境である。一般に、何らかのプログラムを書くためには、(1)どのようなことをコンピューターにさせるのかという、タスク（課題）の解析、(2)ソースプログラムファイルおよびオブジェクトファイルの作成、そして(3)プログラムをrunさせたのon-lineでのデバッギング等々の過程が必要となる。

タスクの解析は、どのプログラム言語を用いる場合にもかかわらず時間を使わなければならぬ部分であるが、機械語を用いる際には、人がコンピューターのCPUとなって、プログラムの1ステップずつを決定し、チェックしなければならない。そしてその際、機上で作成されたプログラムをコンピューターの上で実際に走らせるためのソフトウェアのサポートがどの程度あるのかがプログラム完成までの所要時間の長短の決め手となる。

TK-85では何のサポートもなく、人間が自らの手で命令（instruction）を16進数に変換して、実行可能なファイルを作らなければならない。一方、PC8801では、エディター、アセンブラ、デバッガー等々を含むDUAD-88D（ASCII社製）によるサポートがあった。しかしながら、いずれにしろ、ソースプログラムをコンピューターに打ち込む時にはちょっとしたミスや、ハードアセンプルをする必要が生じた後のオペレーションコードの単純な書き直しが等々が、時としてプログラマーに致命的な打撃を与えることがある。また、プログラム上のエラーを探し出してこれを訂正するデバッギングの過程も、機械語となると非常に苦痛である。「あやまちは人の常（To err is human）」と承知の上で、その人に完全さを要求するとすれば、それはその人に無限とも思える負担をかけていることに他ならない。

最後に、第3の問題点として、マイコンのハードウェアそのものがシステム開発者を苦しめることを述べなければならない。筆者はスキーを箱の制御シス
テムを2度にわたってversion-upしている。すなわち、より実験制御に適したハードウェアを有するマイコンを求めて、TK-85からPC 8801、更にはStaffへと機種を代えてきたのである（Table 1参照）。

TK-85は、そのままではシャーシに収納されていないむき出しのマイコンで、ファイルはカセットテープに落とすしかなく、フロッピーディスクはない、CRTはない、ASCIIあるいはJISのキーボードもない、BASICはまずい等々、ないない尽しで使い勝手は物阻のマイコンであった。

続くPC 8801は、TK-85の欠点の全てをカバーしており、8インチのフロッピーディスク、CRT、キーボード、N88 BASIC等々が利用可能であった。このマイコンを購入した当初は、機械語によるプログラミングはもうやめて、CP/Mの管理下でMBASICを用いてプログラムを書き、BASCOMでコンパイルして実験に供する、あるいは同じくCP/Mの管理下でPASCAL MTやBDS Cなどのコンパイラ型言語を用いて制御を行おうとしたが、割込みおよびCRT画面等々のハードウェア上の問題から、その全てを断念したのである。

割込みに関しては、ユーザーに3レベルが開放されているが、システム内でRS232Cの受信割込み、画面終了割込み等々が行われており、かなり複雑な様相を呈している。さらに、Micro Software Associates 社がリリースしているPC 8801用のCP/Mでは、original CP/Mとは全く異なり、000016番地から010016番地までに存在している、割込み時のジャンプ・テーブルの全てに飛び先の番地が記入されており、ユーザーが入り込む余地がないのである。

一方、CRTについては、画面に書き出すべきデータをDMA（Direct Memory Access）によってCRTIC（CRT Controller）に転送しているが、DMA中はCPUがHALTしている。もしDMA中に入込みがよかったとすれば、割込み信号はラッチ（latch）され、HALTが解けた時に割込みが発生するようにになっているが、時間的なずれ（time lag）が生じる可能性は絶えずある。

他には、モニターおよびN88 BASICとのかね合いで、機械語プログラムを
走らせるためのメモリーエリアが C 00016 番地以降にしかとれず、実験データをメモリー上にあまり多く置いておけないことや、フロッピーディスクドライブ (NEC 純製；FD 1165 AV) のヘッド部分が絶えずフロッピーディスク上に下りており、長時間使用するとフロッピーディスクを壊してしまう可能性があること等々が挙げられる。

IV 理想の実験制御システムについて

ハードウェア・ソフトウェアともに、実験制御に適した理想的マイコン・システムはないのであろうか？筆者はこの問いに対してかつては否定的であったが、幸いなことに、1985年の12月から理想的とも言えるハードウェアとソフトウェアのもとでスキャナー箱を対象とした実験制御システムを作り上げる機会が與えられたのである。

ハードウェアは、札幌にある B・U・G 社が発売している Staff という名称の工業用コンピューターシステムである (CPU: Z-80A)。その構成は PC 8801とほぼ同じであるが、今まで使用してきたマイコンの欠点のほとんど全てを補って余りあるものである。その長所は、以下の6点に要約できる。(1)80系のマイコンの標準的 OS である CP/M の管理下でシステムが走る。(2)割込み関係は original CP/M と同じで、RST 7 以外はユーザーに開放されており、多数のプログラムによって簡単に制御できる。(3) CRT へのデータの転送は RS 232 C を介して 9800 bps (bit per second) で行うことが可能であり、CPU の負担が小さい。(4)メインメモリは、TPA (Transient Program Area) の先頭番地である 010016 番地から CP/M の BDOS の先頭番地の手前まで、完全にユーザーに開放されている。(5)フロッピーディスクドライブは PC 8801 のそれと同じ FD 1165 AV であるが、データの READ/WRITE 時以外はヘッドは OFF しており、フロッピーディスクが壊れる可能性は低い。(6)ハードウェアを構成している回路は、100％公開されている。自作のインターフェイス回路を収容するために10個のスロットも用意されている。
このような理想的なハードウェア上で走らせるソフトウェアとして、2種類のものを考えている。1つは、現在までに蓄積してきたソフトウェアを Staffに移植して、実験者がキーボードから“run”と入力すれば実験が開始されるというように、実験の自動化を目ざす方向である。既存のソフトウェアの移植はほとんど終わっている。

もう1つのソフトウェアは、上述のready-madeのソフトウェアではサポートできないような実験を行う時に使用されるもので、CP/Mの管理のもとに80系のマイコン上で走るPSIである。これこそまさに筆者をして機械語から解放せしめてくれるもので、久保田新氏（日本ロジック研究所）により開発され、その後、坂上貴之氏（慶應義塾大学）との協同作業により現在のバージョンにまで改訂されてきた実験制御システムである（久保田・坂上・堀・田中，1981 a，b）。このシステムは目的Staffに移植中であり、詳しい内容とその評価は他に譲らなければならないが、以下に、久保田ら（1981 a， b）を参考にして、システムの概要を記述することにしよう。

PSIは、実験制御のためにミニコンピューター上で開発されたSKED（Snapper & Kadden, 1973）、ACT（Millenson, 1973）、そしてかつて我々の研究室で開発されたTYMES（滝川, 1978; Takigawa & Mino, 1982）と軌を一にするものであり、状態図（state diagram）に基づいて実験制御のプログラムを作成する。状態図は、時間軸上的ある一点における実験の状態（state）と状態間の移行から成り、実験手続の記述には好都合である。

Fig. 5 の左側は、状態図の考え方を明らかにするために強化スケジュールの1つであるFixed Interval60秒（以下、FI 60”と略記）を例にとった場合を示している。このスケジュールでは、先行の強化反応(reinforced response)から60秒経過後の初発反応に随伴して強化子が与えられるわけではないが、図には5つのstateが記述されている。実験がスタートして、state 0から1に移ってちょ 60秒たと、強化子獲得可能期間(reinforcement available period)に突入したことを示すstate 2 に移行する。もしこのstateにいる時に反応(R)が生起すると、強化子を提示するための装置をONするstate 3 に移行
Fig. 5 Fixed Interval 60秒（FI 60”）を例にとった場合の PSI システムによるプログラム作成（久保田ら、1981 a, b）。
左側のパネル：状態図による実験手順の記述
右側のパネル：PSI 命令によるソースプログラムファイル

する。そして state 3 で 5 秒間が経過すると再び state 1 に戻り、次の60秒間の計測が始まるのである。なお、state 0' は、反応数を数えるためのカウンターに対応している。

実際にプログラムする際には、まず、状態図に基づいて CP/M の ED（エディター）等によりソースプログラムファイルが作成される。その際に重要なことは、PSI 命令という特殊な言語が用いられる点である。Fig. 5 の右側は、PSI 命令による FI 60”スケジュールの state 1 から 3 までのプログラムの実際を示している。図の中で、RESERVE であるのは、次に移行すべき state を指定する命令である。わかりにくい命令はこれくらいで、他は TIMER・ON・OFF 等々、実験者になじみのある表現が用いられている。状態図さえ描ければ実験の制御プログラムを書けるわけで、機械語で 1 ステップずつプログラムを書いていくのとは天と地ほどの隔りがある。

作成されたソースファイルは、CP/M のもとで、マクロアセンプラープロ－ン（MAC）と PSI システム内のライブラリー（PSI/L）によりアセンプルされて、最終的に OS のもとで即行可能な COM ファイルが生成される。実際にこのファイルが run される時には、メモリー上の 010016 から 200016 番地に常駐するモニター（PSI/M）によって管理されるのである。

もし PSI が種々の実験において十分にその能力を発揮できるものであれば、心理学者は実験制御に伴う思いつきから解放され、より自由に、またより生産
的に研究を展開してゆけるはずである。

Ⅴ 結びにかえて

過去何年間かのマイコンとの関わりを振り返って見てきたわけであるが、なんだという遠回りをしてきたのだろうかという思いとともに、それでもなおより良い実験環境を求め続けてきたことを覚えさせられる。コンピューターは道具である、そのことをもう一度確認して、現在与えられている環境の中で研究に精を出してゆきたい。

マイコンを始めとする種々の機器に触れることを可能にして下さり、また論文執筆の機会を与えて下さった関西学院大学文学部教授今田寬先生に感謝申し上げます。また、マイコンに関する know-how を獲得する過程でお世話になった関西学院大学大学院研究員黒瀬哲郎先生、武田薬品中央研究所山崎直樹先生に感謝致します。さらに PSI を譲渡して下さいました、日本コンシュ研究所久保田新先生、慶応義塾大学坂上貢之先生に謝意を表したいと存じます。

註
(2) 割込みとは、現在実行中の処理を中断して予め定められている他の処理を行うことをさす。割込みはハードウェア的にもソフトウェア的にもかけることができる。
(3) CP/M は、80系の標準的オペレーティング・システムであり、米国ディジタル・リサーチ社製である。MBASIC, BASCOM, PASCAL MT*, BDS C は、いずれも CP/M 上で走るソフトウェアプロダクトである。
(4) メインメモリーと周辺機器の間で、CPU を介さずに直接にデータのやり取りを行うことす。

参考文献
藤 健一 1982 心理学の実験制御とマイクロコンピュータ 立命館大学, 439-441, 1-31
久保田新・坂上貢之・堀 耕治・田中 敦 1981a マイクロコンピューターによる行動実
心理学実験におけるマイクロ・コンピューターの役割について

駒野オンラインシステム 1. リアルタイムマルチプルタイマー 日本心理学会第45回大会発表論文集, p. 9
久保田新・坂上貴之・塚谷耕治・田中毅 1981b マイクロコンピューターによる行動実験用オンラインシステム 2. プログラミング言語 日本心理学会第45回大会発表論文集, p. 10
美濃哲郎 1980 心理学研究と電子計算機 関西学院大学人文論究 30, 46-67
中谷和夫 1981 心理学におけるマイクロコンピューター利用の現状と展望 サイコロジー 16, 12-17
斎藤洋典 1983 マイクロ・コンピューターによる3チャンネル・プロジェクト式タキストスコープ制御及び反応時間計測システム 関西学院大学人文論究 33, 74-93
瀧川哲夫 1978 実験制御モニター TYMES/PTS 親和女子大学研究論叢第11号
吉村浩一 1981. 我が国の「心理学におけるマイクロコンピュータ利用状況」についての調査結果から 人文 27, 1-27