



Title	FORTTRANによる画像入出力ボードの制御
Author(s)	牛坂, 健
Citation	北海道大学電子科学研究所技術部技術研究報告集, 3, 67-70
Issue Date	1996-03-01
Doc URL	https://hdl.handle.net/2115/1464
Type	departmental bulletin paper
File Information	KJ00000697046.pdf



資料

FORTRANによる 画像入出力ボードの制御

牛 坂 健

パソコンの拡張スロットに挿入して使用する画像入出力ボードは、テレビカメラなどの出力画像信号をデジタル化してパソコンの記憶装置に保存する装置として比較的安価に市販されています。

私の所属する研究室（光システム計測研究分野）でも画像入出力ボードを画像処理を伴う研究に使用していますが、画像入出力ボードの制御と画像処理のための計算をプログラム言語の「FORTRAN」で行っています。

私が製作したこのソフトウェアは、大変珍しいことに当研究室の多くの方々に利用して頂いているので（目的の研究が終了すると忘れ去られてしまうのが普通なのです！）、この画像処理プログラムについて簡単に紹介させていただきます。

使用している画像入出力ボードは、NEC製パソコン「PC-9801」の拡張スロットに挿入して使用するフォトン社製の「イメージデジタイザ FRM2-512」という画像入出力ボードです（以下ではFRMと略記する）。本ボードはPC-9801の増設RAM領域 80000h - 9FFFFh にボードに実装されている画像メモリーを割り付ける構造になっており、このためコンピュータの主メモリーを 640KB 確保する時は、画像メモリーとコンピュータの主メモリーの上位 128KB が増設RAM領域で重なるので、バンクメモリーと併用してメモリーを切り替えて使用する必要があります。バンクメモリーは最低 128KB の容量のものがが必要です。なお、画像メモリーは 512 x 512 画素の白黒画面を 2 画面もっています（図 1）。

ボードの制御は、3つの入出力ポートを介して行われ、画像データの取り込み、再表示、直線性補正テーブル（ルックアップテーブル）のオン/オフ、直線性補正データの書き込みなどの制御を

します。また、画像メモリーとバンクメモリーの切り替えは I/O アドレス 00ECh にバンク番号を書き込むことで行われます。画像メモリーのバンク番号は F0h ~ FFh の範囲で変更可能です。

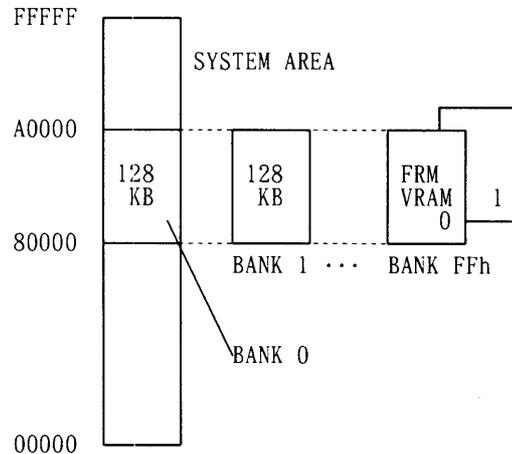


図1 メモリマップ

FRMの制御と簡単な画像処理を行うためのソフトウェアは既に述べたようにフォートラン言語で作成しました。本プログラムはフォートラン副プログラム（サブルーチン）からなり、使用者がその副プログラムを利用して研究目的に適った画像処理プログラムを作成できるようになっています。このソフトウェアを利用することで、利用者はハードウェアを意識することなく、画像入出力ボードを使用できます。

以下で説明するソフトウェアは、研究室（光システム計測研究分野）の標準フォートランである Pro-FORTRAN77 に用いるサブルーチン副プログラムです。この資料末に副プログラム名とその機能を一覧にまとめてあります。副プログラム名は、FRMに付属してきたソフトウェアや他の画像処理ソフトを参考にして決定しました。

サブルーチン一覧から分かるように、おおまかには次のような目的のプログラムから成っています。

- ・FRMの画像取り込み、表示などの制御
- ・画像メモリーからプログラム内配列変数への画像データのコピー

- ・二つの画像メモリー間の簡単な演算
(加算、減算、AND、OR、XORなど)
- ・画像データのファイルへの書き込み及び
ファイルからの読み出し

以下に使用法を説明します。

最初に必ず FMINIT(208, 255) をコールします。二つのパラメータはボードの設定によりますが、出荷時の設定のままの時は(208, 255)です。これによって、FRMの制御に必要なパラメータや変数がプログラム内にセットされます。これらのパラメータや変数はCOMMON宣言されていて、プログラムの実行中はメモリ上の共通領域におかれ、これらのパラメータや変数を必要とする副プログラムがこの共通領域のデータを読んで動作するようになっています。大容量のバンクメモリを実装している場合は、空いているバンクメモリを画像データの一時保存場所として使用できるようにしました。2面あるFRMのVRAMは2面として使用することの他に、片面を画像処理用バッファメモリとしても使用できます。

必要ならプログラム内で画像データバッファとして2バイトの整数型2次元配列変数を宣言します。この配列変数はコンピュータのメモリの制限から300 X 300程度が上限です。画像データは1バイト(8ビット)なのでINTEGER*1のデータタイプを使用すると512 X 512画素の1画面分の配列変数を確保できますが、1バイト整数は-127 ~ 127の値しかとれませんので初心者には誤解しやすいと言うことでINTEGER*2を使用しています。

画像データは、データファイルとコントロールファイルの二つのファイルで保存することとしました。画像データと画面との関係は図2のようになります。すなわち、画面左上の座標を(0, 0)とし右下の座標を(511, 511)としています。

ファイルに保存する画像データは、

$$\begin{matrix} (0, 0) \cdots (511, 0), \\ (0, 1) \cdots (511, 1), \\ \vdots \\ (0, 510) \cdots (511, 510), \\ (0, 511) \cdots (511, 511) \end{matrix}$$

の順で、画素データを1バイトの文字型に変換してランダムファイル形式として保存します。ランダムファイルですので、一行単位(画面の1ライン単位)で直接にファイルの読み書きができます。画面全面のデータを保存すると容量が約256KBのファイルが作成されます。

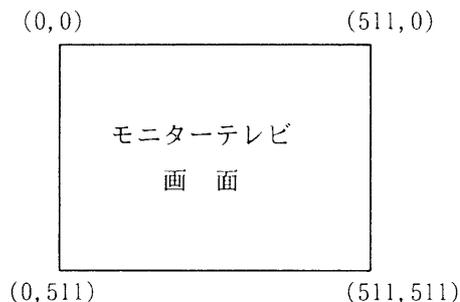


図2 画面座標

保存する画面範囲を自由に指定でき、その画面範囲を示す座標をコントロール・ファイルに記録します。コントロール・ファイルは拡張子を.CTLとしてデータファイルと区別し、ファイルの形式は、下記のように1バイトの文字型データ12文字で保存領域の画面上の座標をシーケンシャルファイルの形式で記録することとしました。

3A1	3A1	3A1	3A1
始点座標		終点座標	

以上、簡単にフォートラン言語で画像入出力ボードを制御するプログラムの紹介をしました。本資料末に、サブルーチン名とその機能を示しましたので、使用法はお分かりいただけることと思います。本プログラムは、初期のものは、画像入出力ボードの制御ポートのアクセスや画像メモリの読み書きの部分を除き、ほとんどフォートラン言語で書かれていましたので動作が遅く、特にCPUが8086の機種では実用になるような速度ではありませんでした。その後、大部分をアセンブラ言語で書き直しましたので、動作速度は大幅に向上しています。

画像入出力ボード制御用サブルーチン一覧

FMINIT (fmport, bankad)

FRMのコントロールに必要な変数のデフォルト値を与える。
プログラムの最初で必ずこのサブルーチンをコールしてください。
引数 fmport, bankad はいずれも integer*4 の正数で与えます。
fmport : Control port address of FRM2-512.
bankad : Bank number of memory of FRM2-512
工場出荷時の設定は fmport=208, bankad=255 です。
ボードの設定を変更したときは、適切な数値を与えてください。

APAGE (ipage)

画面のページ設定と問い合わせ。
設定 1または2を与える。 1=A画面, 2=B画面
ipage のデータが不適の時は何もせずに、ipage=-1としてリターンします。
問い合わせ ipage=0 の時は、現在の画面のページの問い合わせで、戻り時 ipage にページ番号(A=1, B=2)がセットされます。

IGET

画像信号を画面Bに1フレーム取り込む。
戻り時はFRMバンクが0で画面B表示になります。

IVGET

画像信号を画面Aに1フレーム取り込み、内部同期とする。
戻り時はFRMバンクが0で画面A表示になります。

IVGETB

画像信号を画面Bに1フレーム取り込み、内部同期とする。
戻り時はFRMバンクが0で画面B表示になります。

ISURU

入力画像信号を直接モニターTVに表示する。
戻り時はFRMバンクが0で画面A指定となります。

ISURUB

入力画像信号を直接モニターTVに表示する。
戻り時はFRMバンクが0で画面B指定となります。

FRZ

画面Aのメモリ像をビデオモニターに表示する。

FRZB

画面Bのメモリ像をビデオモニターに表示する。

ICHNG

画面Aのデータと画面Bのデータを交換する。
画面は常にAになります。

ICPYAB

画面Aのデータを画面Bにコピーします。
実行終了後画面はBになります。

ICPYBA

画面Bのデータを画面Aにコピーします。
画面は常にAになります。

IGAND

画面Aと画面BのAND演算をし画面Aに表示します。

IGOR

画面Aと画面BのOR演算をし画面Aに表示します。

IGXOR

画面Aと画面BのXOR演算をし画面Aに表示します。

IGADD

画面Aと画面Bを加算し画面Aに表示します。
濃度データが255を越えるときは255にします。

IABAV

画面Aと画面Bの平均をとり画面Aに表示します。
すなわち、 $([A] + [B]) / 2 \rightarrow [A]$

IGSUB1

画面Aから画面Bを減算し画面Aに表示します。
濃度データが負になるときは0とします。

I G S U B 2

画面Bから画面Aを減算し画面Bに表示します。
濃度データが負になるときは0とします。

I C L R (n)

現在の画面を任意の明るさ n (0 から 2 5 5) でクリヤする。

I T W O (n)

画像データの2値化。境界値 n (0 から 2 5 5) を指定します。

I N E G

現在の画面の濃度を反転する。

I B A N K (n)

1 画面内のバンク n (0 - 3) を指定する。
バンクの設定以外の動作モードの変化はありません。

B P E E K (i x , i y , i d a t a)

任意のスクリーン座標の濃度データを読み取る。
(i x , i y) : 画面座標 0 - 5 1 1
i d a t a : 濃度データ 0 - 2 5 5
動作モードの変化はありません。

B P O K E (i x , i y , i d a t a)

任意のスクリーン座標に濃度データを書き込む。
(i x , i y) : 画面座標 0 - 5 1 1
i d a t a : 濃度データ 0 - 2 5 5
動作モードの変化はありません。

B P E E K L (i x , i y , i d a t a , n)

スクリーン座標 (x , y) から n 個のデータを読む。
i x , i y : 画面座標
i d a t a (n) : 出力 1 次元配列

B P O K E L (i x , i y , i d a t a , n)

スクリーン座標 (x , y) から n 個のデータを書き込む。
i x , i y : 画面座標
i d a t a (n) : 書き込むデータを格納した 1 次元配列

C P E E K L (i x , i y , c d a t a , n)

スクリーン座標 (x , y) から n 個のデータを読む。
i x , i y : 画面座標
c d a t a (n) : 出力 1 次元配列 character*1 type
B P E E K L と出力データ型が異なるだけです。

C P O K E L (i x , i y , c d a t a , n)

スクリーン座標 (x , y) から n 個のデータを書き込む。
i x , i y : 画面座標
c d a t a (n) : 書き込むデータを格納した 1 次元配列
character*1 type
B P O K E L と入力データ型が異なるだけです。

L T I N I T (l n)

ルックアップテーブルの初期化。
l n : ルックアップテーブルの番号 (0 - 7)
ルックアップテーブル書き込み用カウンタを 0 にし、指定のテーブルを
選択する。

L T P U T (i d a t a)

ルックアップテーブルに 1 個のデータを書き込む。
一度の書き込みで書き込み用カウンタが 1 だけ増加します。256 回書き込むとカウンタは零
に戻ります。ルックアップテーブルは予め LTINIT 選択しておきます。

L T O F F

ルックアップテーブルを休止状態にします。

L T O N

ルックアップテーブルを動作状態にします。

C U R O N

カーソル機能を ON にする。