

Addressing Ten Fingers(2)  
\*\*\* a tool as a substitute for human voice \*\*\*

Manaka Shunyuu

Abstract :

We try to express "50 ON" using the body parts namely finger, arm, tongue, eyelid, mouth, face, neck and eyeball for address specification. We call the way body sign. Because both body sign and artificial voice which is generated from the algorithm of body sign have perfect compatibility on voice of a person who has normal vocal function, they can be applied to stuttering treatment.

## 1. The primates

### 1.1. State of fingers

A finger number is expressed with contact between thumb and palm or between fingers considering medical characteristic of finger motion and noticeability.

- finger number 1 : contact between thumb and palm
- finger number 2 : contact between thumb and index finger
- finger number 3 : finger number 2 + contact between index finger and middle finger
- finger number 4 : finger number 3 + contact between middle finger and third finger
- finger number 5 : finger number 4 + contact between third finger and little finger

The following is the other states.

- release : no finger contact
- 1st goo : contact between thumb and index finger + contact between the other fingers and palm
- 2nd goo : contact between fingers except thumb and palm

### 1.2. Both hands character

In address type finger character, a state of fingers means mainly a number as an address differing from a hieroglyphic type one. On left fingers, finger number 1~5 express あ line~な line.

- あ line~な line : left finger number 1~5

If left hand is shifted, the number increases by 5. It means one of は line~わ line.

- は line~わ line : left hand shift  $\Rightarrow$  finger number 6~10
- elements of わ line : わ、を、ん、 、?

The direction of the left hand shift for the lines is  $\uparrow$ .

- The direction of the left hand shift :  $\uparrow$

If we show the palm of released left hand, it means は line.

- release of left hand : は line

On right fingers, finger number 1~5 express vowels a, i, u, e, o.

- vowels a, i, u, e, o : right finger number 1~5

The following is right hand shift on voiced sound, p-sound, small character.

- voiced sound : horizontal(right or left) shift in finger number 1~5

- ・ p-sound : up shift in finger number 1~5
- ・ small character : down shift in finger number 1~5

When we take down shift on small character "っ", the finger number is not limited to 3.

- ・ small character "っ" : down shift in finger number 1~5

We call advance modification action and previous mention action of small character advance small character and previous small character respectively.

- ・ advance small character : down shift of left hand with the hand released
- ・ previous small character : down shift of left hand with the hand finger number 4
- ・ advance small character : down shift of right hand with the hand released

If there are small characters, the way of expression is not unique. The element is down shift of finger number, down shift of released hand, down shift of right hand, down shift of left hand, one down shift, two down shifts, and so on.

- ・ しょっ
- ・ し : left 3、 right 2
- ・ よ : left 8、 right down shift of 5(press)
- ・ っ : left 4、 right down shift of 3(press)
- ・ し : left 3、 right 2
- ・ down shift with right hand released(near press)
- ・ よ : left 8、 right 5
- ・ down shift with right hand released(near press)
- ・ っ : left 4、 right 3
- ・ し : left 3、 right 2
- ・ two down shifts with right hand released(two near presses)
- ・ よ : left 8、 right 5
- ・ っ :
- ・ two down shifts with left hand released(two far presses)
- ・ し : left 3、 right 2
- ・ よ : left 8、 right 5
- ・ っ :

If the number of down shifts is 2, we leave out the ten fingers expression of small character "っ". Down shift of left hand is used only before a large character which precedes a small character.

When only "っ" follows a large character in small character, we leave out the ten fingers expression of small character "っ" if we take down shift with left hand finger number 4.

- ・ しっ
- ・ し : left 3、 right 2
- ・ っ : left 4、 right down shift of 3(press)
- ・ down shift with left hand finger number 4(4-press)
- ・ し : left 3、 right 2
- ・ っ :

Long sound is expressed by horizontal shift before chest of released hand. Question mark is expressed by up shift of left hand in 1st goo too.

- - : space drawing(release)
- ? : up shift of left hand(1st goo)

Backspacing of one character is possible.

- Backspacing of one character : We make a hand 2nd goo, the direction of the thumb in parallel with the horizontal and the hand move horizontally.

function animal	left hand	right hand	line shift	voiced sound	p-sound	small character	long sound
premates	1,2,3,4,5	1,2,3,4,5	left ↑	right ← or →	right ↑	right ↓	space drawing

function animal	あ line			Question mark	advance small character
premates	left(release)			left ↑(1st goo)	left ↓(release), right ↓(release)

function animal					previous small character "っ"
premates					left ↓(finger number 4)

### 1.3. Single hand character

If the number of hands which are movable is 1, we make it function as the other hand too. For the sake of ease, we assume that only left hand is movable. First, we arrange it around left shoulder one-dimensionally and choose one of あ line~わ line.

- left hand : line choice

We move it right with it released and arrange it around neck one-dimensionally.

- left hand : right movement

We choose one of vowels a, i, u, e, o.

- left hand : vowel choice

The following is left hand shift on voiced sound, p-sound, small character.

- voiced sound : horizontal(right or left) shift in finger number 1~5
- p-sound : up shift in finger number 1~5
- small charcter "っ" : down shift in finger number 1~5
- advance small character : down shift of left hand with the hand released
- previous small character : down shift of left hand with the hand finger number 4
- advance small character : down shift of left hand with the hand released after right movement of it

The others are the same as both hands character.

function animal	left hand	right hand	line shift	voiced sound	p-sound	small character	long sound
premates	1,2,3,4,5	1,2,3,4,5	left ↑	right ← or →	right ↑	right ↓	space drawing

function animal	あ line			Question mark	advance small character	separator
premates	left(release)			left ↑(1st goo)	left ↓(release), right ↓(release)	right movement of left hand



- small character : both half-opened
- ー : next to や
- Backspacing of one character : next to ㇿ

Voiced sound, p-sound and small character, all is advance modification action.

function	eye number	vowel	line shift				long sound
animal							
premates	eyeball:left,right,centre,upper,lower	right closed	left closed				next to や

function	separator						
animal							
premates							left double-click,right double-click

function	voiced sound			p-sound			small character	
animal								
premates			right half-opened with left closed			left half-opened with right closed		both half-opened

## 2. The other animals

If there is a body part which corresponds to a hand of the premates in the other animals, the learning will continue progressively. In a dog and a horse which have four legs, though the balance is disturbed a bit, a leg can be used for character expression. An elephant can move the trunk with complete control like a hand of the premates. The trunk is unique, and so speedy expression is not expected. However, the range of movement of the trunk tip is large, and so certainty is expected.

- dog, horse

function	left body number(left leg)	right body number(right leg)	line shift				
animal							
dog,horse	left,right:1,2;left,centre,right in front:3,4,5	the same as left leg	face left;tail left				

function	voiced sound	p-sound	small character	long sound
animal				
dog,horse	face right	looking up	looking down	mouth opened

- elephant

function	body number(trunk)	line shift				
animal						
elephant	left,right in front and lower level:1,2;left,centre,right in front and upper level:3,4,5	left ear				

function	voiced sound	p-sound	small character	long sound	separator
animal					
elephant	right inner part	near head	inner part	mouth opened	tail(left,right)

- dolphin, sea lion

function	body number	line shift				
animal						
dolphin,sea lion	1:tongue tip left,2:tongue tip right,3:tongue tip left→tongue tip right 4:left closed,5:right closed	face left				

function	voiced sound	p-sound	small character	long sound	separator
animal					
dolphin,sea lion	face right	looking up	looking down	mouth opened	front fin(left,right)

- bird

function animal	body number(left leg/right leg)	line shift	voiced sound	p-sound	small character		
bird	left,right,front:1,2,3 left,right:4,5	face left	face right	looking up	looking down		

function animal							long sound	separator
bird							mouth opened	wing(left,right)

### 3. Language education of animal

There is a possibility that the learning will stagnate except the case that an animal is excellent especially and the number of directors is larger than 1 in body sign. Body number is accompanied by uncertainty in the recognition of the position if the empty air or the ground is pointed. There is a possibility of misrecognition in the change of the face. There is a possibility of oversight in the change of the tail. Considering them, we introduce a way in which learning speed is slow, but teaching and understanding are easy. In an elephant and a chimpanzee, we use the group of symbols which is designed like Figure 1.

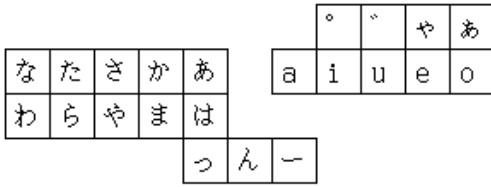


Figure 1

This is an arrangement figure of keys of a keyboard. We call an object which is painted really like it graphic keyboard. A director puts up and reads out combined kana characters and teaches symbols to touch which corresponds to the characters. If there is a mistouch, retouch is urged. We call it large scale direction. It gets near to the last keyboard if the following devices are attached.

- touch sensor
- beep
- LED(bicolor or two for each symbol)
- PC, Speaker

First, Oyatsu is used in order that we get an animal to understand combination of symbols and arrangement about time. However, an animal must be able to obey "mate". If an animal becomes proficient, we feed it after a lesson.

We assume the word which an animal learns first to be oyatsu. Oyatsu is not a name of specified food but one of forms of meal and a generic name of foods which we eat at the time. We get an animal to master "50 ON" with learning a name of an element of oyatsu. For example

- あんず
- かき
- すいか
- トマト
- なし
- バナナ
- もも
- ゆりね
- りんご
- わかめ

If all the "50 ON" are mastered, we get an animal to master small character.

- ちしゃ
- きゅうり
- みょうが
- ビッツァ
- キウイ

- ・ トウラウト
- ・ チェリー
- ・ ウオルナット

However, we except feed-inhibited foods on an animal. Because the weight of loan word is large in small characters of あ line, the number of characters tends to become great. We recommend that beginning short words, if an animal becomes proficient, long words are put in hand.

The following is an example of order of oyatsu.

- ・ もも、かき、なし、すもも、トマト、ゆりね、あんず、いちご、バナナ、パイナップル、プルーン

The following is a procedure for teaching もも.

- (1)reads out もも putting up a picture of もも and a board on which もも is written
- (2)puts a piece of もも to ま of graphic keyboard
- (3)urges a touch on ま
- (4)gives the piece if touches it
- (5)puts a piece of もも to vowel 'o' of graphic keyboard
- (6)urges a touch on 'o'
- (7)reads out も and gives the piece if touches it
- (8)puts a piece of もも to ま of graphic keyboard
- (9)urges a touch on ま
- (10)gives the piece if touches it
- (11)puts a piece of もも to vowel 'o' of graphic keyboard
- (12)urges a touch on 'o'
- (13)reads out も and gives the piece if touches it

This piece feed can be cleared without any difficulty if the same routine is done in an usual oyatsu.

The following is a procedure for teaching かき.

- (1)reads out かき putting up a picture of かき and a board on which かき is written
- (2)puts a piece of かき to か of graphic keyboard
- (3)urges a touch on か
- (4)gives the piece if touches it
- (5)puts a piece of かき to vowel 'a' of graphic keyboard
- (6)urges a touch on 'a'
- (7)reads out か and gives the piece if touches it
- (8)puts a piece of かき to か of graphic keyboard
- (9)urges a touch on か
- (10)gives the piece if touches it
- (11)puts a piece of かき to vowel 'i' of graphic keyboard
- (12)urges a touch on 'i'
- (13)reads out き and gives the piece if touches it

When ”ゝ”, ”°”, ”っ”, ”ん” and ”ー” appear, we do piece feeds of them before one of vowel.

When a small character of や line or a small character of あ line appears, we do the job on vowel two times in succession. For example, in きゅ

- (2)puts a piece to か of graphic keyboard
- (3)urges a touch on か
- (4)gives the piece if touches it

(5)puts a piece to small character や of graphic keyboard

(6)urges a touch on や

(7)gives the piece if touches it

(8)puts a piece to vowel 'i' of graphic keyboard

(9)urges a touch on 'i'

(10)gives the piece if touches it

(11)puts a piece to vowel 'u' of graphic keyboard

(12)urges a touch on 'u'

(13)reads out きゅ and gives the piece if touches it

If lead of directors is gone, an animal may not be able to do the continuous touch on vowel. In this case, we can substitute a large character for it. For example

・ きゅうり

If the above basis is mastered, we move on to learning of realistic names.

・ あなた、わたし

あなた is a director whom an animal faces. The director teaches あなた with voice and writing on a blackboard pointing self. Because わたし is the animal, taking a picture of the animal in the situation, pointing the animal, the director teaches わたし with voice, writing on a blackboard and the picture pointing the animal. The picture is not necessarily needed, however, we add it in the meaning of insurance. However, there is a possibility that the animal mistakes あなた and わたし for humans and animals respectively.

About time, first, we teach night and day.

・ よる、ひる

About them, two pictures of them serve the understanding.

Making a time axis in which night and day are arranged for yesterday, today, tomorrow, we attach a picture of the director and the animal to today and a yesterday's evidential picture to yesterday. We assume the coordinate name of the time axis to be じかん.

・ きのう、きょう、あした、じかん

We teach words which resemble these too.

・ まえ、いま、あと

In this case, we use another time axis. Assuming one point to be now, we attach a picture of the director and the animal to it. We assume the past and the future to be a half beeline which corresponds to < and a half line which corresponds to > respectively. We give different colors or tones to them.

#### 4. Stuttering treatment

We try to apply body sign to stuttering treatment. When an utterance is incomplete on specific words, the question seems to be solved relatively easily.

・ extension : わーわたしは

・ succession : わわわたしは

・ blocking : ... わたしは

In treatment utterance, a client moves on to covering by body sign or artificial voice at once if a client feels a crisis.

・ boody sign(ten fingers expression)



- ・ artificial voice(add10)

For example, blocking. ”, ” is an end of body sign or artificial voice.

- ・ わ, たしは
- ・ わたしは,

I do not know the details of blocking, and so we consider the latter case that the covering is needed for 4 characters. In the first covering, a client closes the mouth on all the four characters.

- ・ ; ; わたしは,

After the first covering, we replace the closing of the mouse with a vocal action.

- ・ ; わ; たしは,
- ・ ; わた; しは,
- ・ ; わたし; は,
- ・ ; わたしは; ,

’; ’ is both ends of a vocal action section.

In reverse order

- ・ わたしは; ; ,
- ・ わたし; は; ,
- ・ わた; しは; ,
- ・ わ; たしは; ,
- ・ ; わたしは; ,

The kind of vocal action is no sound vocalization, oral cavity vocalization, vocal cords vocalization. We can interpret body sign and artificial voice as a kind of pacemaker, duet partner.

Before the covering, we do reading of addressing ten fingers fully on the three vocal actions.

- ・ addressing ten fingers : body sign、 artificial voice(”50 ON”+voiced sound+p-sound+ん+small character)
- ・ reading : no sound vocalization, oral cavity vocalization, vocal cords vocalization

We may limit the addition of ん and small character to the kana combination of the case concerned. If we can replace わ in critical situation with the reading in a perfect form conditioned reflexively, a client will utter being accompanied by body sign for the time being in general social life. I suppose that extension, succession < blocking in the probability of success.

## 5. Supplement

The best positions of 無変換 key and ひらがな・カタカナ key in 106 are as follows:

- ・ 無変換 key : overlaps with F by half key in a backslash type
- ・ ひらがな・カタカナ key : overlaps with L by half key in a slash type

The following is a table of the last spec.

function	な line	た line	さ line	か line	あ line	a	i	u	e	o
Keyboard										
106	A	S	D	F	C	.	L	;	:	]

function	っ	ー	ん	line shift	p-sound	voiced sound	”や” line	”あ” line
Keyboard								
106	無変換	Space	変換	ひらがな・カタカナ	O	P	@	[

We can use a keyboard except 106 if we make key assignment flexible. When we try to use thumb shift keyboard, it is the condition that the keysym(XK\_~) is.

function \ Keyboard	な line	た line	さ line	か line	あ line	a	i	u	e	o
Fujitsu thumb shift	A	S	D	F	C	M	J	K	L	;

function \ Keyboard	っ	ー	ん	line shift	p-sound	voiced sound	”や” line	”あ” line
Fujitsu thumb shift	無変換	thumb left	変換	thumb right	U	I	O	P

If we have a knowledge of PC's keyboard, we can alter 106 in imitation of thumb shift keyboard. The following is the corresponding relations.

- thumb left  $\iff$  Space
- thumb right  $\iff$  right Ctrl

```
no.25::0    /* ideal */
old:else if(!BPideal && GKS(R11)<0) shift=1;
new:else if(!BPideal && GKS(XK_Ctrl_R)<0) shift=1;
```

The modification parts on 101 in the source code are R6, R11 in `get_type()` and L11, L6, R4, R5, R10 of upper type.

```
/* get_type(), else{}, left:106, right:101 */
· R6=XK_Henkan  $\implies$  R6='m'
· R11=XK_Hiragana_Katakana  $\implies$  R11=','

/* get_type(), upper type, left:106, right:101 */
· L11=XK_Muhenkan  $\implies$  L11='c'
· L6=' '  $\implies$  L6='v'
· R4='@'  $\implies$  R4='['
· R5='['  $\implies$  R5=']'
· R10='^'  $\implies$  R10='='
```

Because we use 101 as upper type, the leading character of executable file must be '@'.

- @add10

function \ Keyboard	な line	た line	さ line	か line	あ line	a	i	u	e	o
101	Q	W	E	R	F	L	O	P	[	]

function \ Keyboard	っ	ー	ん	line shift	p-sound	voiced sound	”や” line	”あ” line
101	C	V	M	,	9	0	-	=

If we combine 106 with solaris, we can not use the three Japanese keys of the lowest line. In this case, we make a modification which is partly the same as one of 101.

```
no.26::0    /* break */

/* get_type(), else{}, left:106, right:101 */
· R6=XK_Henkan  $\implies$  R6='m'
· R11=XK_Hiragana_Katakana  $\implies$  R11=','
```

```
/* get_type(), upper type, left:106, right:101 */  
· L11=XK_Muhenkan  $\implies$  L11='c'  
· L6=' '  $\implies$  L6='v'  
  
· @add10
```

The last way is also effective.

```
no.25::1 /* ideal */  
· physical swap of wirings
```

In the system of the last Figure 1, we must examine an addition of an infinite long sound closely if there are つ, ん at the end of combined kana characters.

- (1) ”あ” : addition
- (2) ”あー” : addition
- (3) ”あーっ”、 ”あーん” : addition
- (4) ”あっ”、 ”あん” : no addition

In (3), the system makes ー an infinite long sound. Therefore, in order to make combined kana characters a sound just as they are, we must emit sounds of ーっ, ん if an infinite long sound is broken.

- seamless addition of a sound of ーっ : uses a vowel of ー

The font files are made from video bios 1800h in DOS.

- getfont.c : for FreeDOS/V(fdos0138.img), open-watcom-c-dos-1.9.exe needed
- getfont.exe  $\implies$  DOSFONT1.BIN, DOSFONT2.BIN, SJS.BIN
- add10.c : v1, nkf needed, in set\_SB\_DB(), #if /\*UTF8\*/0 into #if UTF8
- add10  $\implies$  sjis.bin, utf8.bin

Addressing Ten Fingers(2)  
\*\*\* 声帯による声の代用としてのツール \*\*\*

間中春由

アブストラクト：

体の部位、すなわち指、腕、舌、脛、口、顔、首、眼球をアドレス指定に用いて 50 音を表現することを試みます。この手法を体話と称します。体話、体話のアルゴリズムで繰り出される人工音声、いずれも健常者の音声に関して完全な互換性を有しているので吃音治療に応用可能です。

1. 霊長類

1.1. 指の状態

指数字は指の運動の医学的特性と視認性を考慮して親指と掌または指と指の接触で表します。

- ・ 指数字 1：親指を掌につける
- ・ 指数字 2：親指と人差し指を接触させる
- ・ 指数字 3：2 の状態で中指を人差し指に接触させる
- ・ 指数字 4：3 の状態で薬指を中指に接触させる
- ・ 指数字 5：4 の状態で子指を薬指に接触させる

指の状態としては指数字の他に以下のようなものがあります。

- ・ 指開放：指の接触がない
- ・ 半グー：親指と人差し指を合わせ、他の指を掌につける
- ・ 四グー：親指以外の指を掌につける

1.2. 両手文字

アドレス型の指文字では象形型のそれとは異なって、指の状態は主にアドレスとしての数字を表します。左手の指は指数字 1～5 があ行～な行を表します。

- ・ あ行～な行：左手の指数字 1～5

左手をシフトすれば指数字は 5 を加えられます。は行～わ行を表します。

- ・ は行～わ行：左手のシフトあり ⇒ 指数字 6～10
- ・ わ行の要素：わ、を、ん、 、？

は行～わ行のための左手のシフトの向きは上とします。

- ・ 左手のシフトの向き：上

指開放の左手の掌を見せればは行とします。

- ・ 左手の指開放：は行

右手の指は指数字 1～5 が母音 a、i、u、e、o を表します。

- ・ 母音 a、i、u、e、o：右手の指数字 1～5

濁音、半濁音、小文字に関する右手のシフトは以下のようになっています。

- ・濁音：指数字1~5で水平(右または左)シフト
- ・半濁音：指数字1~5で上シフト
- ・小文字：指数字1~5で下シフト

小文字”っ”の場合の下シフトでは指数字は3でなくてもかまいません。

- ・小文字”っ”：指数字1~5で下シフト

小文字の前修飾動作、先述動作を各々前小文字、先小文字と称します。

- ・前小文字：左手を指開放として下シフト
- ・先小文字：左手を指数字4として下シフト
- ・前小文字：右手を指開放として下シフト

小文字が伴う場合はその表現の仕方は一通りではありません。その要素は指数字下シフト、指解放下シフト、右手下シフト、左手下シフト、一回下シフト、二回下シフトなどです。

- ・しゅっ

- ・し：左手指数字3、右手指数字2
- ・よ：左手指数字8、右手指数字5で下シフト (press)
- ・っ：左手指数字4、右手指数字3で下シフト (press)

- ・し：左手指数字3、右手指数字2
- ・右手を指開放として下シフト (near press)
- ・よ：左手指数字8、右手指数字5
- ・右手を指開放として下シフト (near press)
- ・っ：左手指数字4、右手指数字3

- ・し：左手指数字3、右手指数字2
- ・右手を指開放として2回下シフト (two near presses)
- ・よ：左手指数字8、右手指数字5
- ・っ：

- ・左手を指開放として2回下シフト (two far presses)
- ・し：左手指数字3、右手指数字2
- ・よ：左手指数字8、右手指数字5
- ・っ：

2回下シフトの場合に限り、小文字”っ”の十指表現を省略します。左手下シフトは小文字を伴う大文字の前で行います。小文字が”っ”だけの場合は左手を指数字4として下シフトすれば小文字”っ”の十指表現を省略します。

- ・しっ

- ・し：左手指数字3、右手指数字2
- ・っ：左手指数字4、右手指数字3で下シフト (press)

- ・左手を指数字4として下シフト (4-press)
- ・し：左手指数字3、右手指数字2
- ・っ：

長音は指解放の手の胸の前の水平シフトで表現します。疑問符は半グーの左手の上シフトでも表現できます。

- ・ー：空間描画 (指解放)
- ・?：左手の上シフト (半グー)

一文字後退削除が可能です。

- ・一文字後退削除：指を四グーにする。親指の方向を水平にし、手を水平移動させる。

機能 動物	左数字	右数字	行シフト	濁音	半濁音	小文字	長音
霊長類	左手 (指数字:1,2,3,4,5)	右手 (同左)	左上	右手水平	右上	右下	空間描画

機能 動物	は行	疑問符	前小文字
霊長類	左手指解放	左上 (半グー)	左下 (指解放)、右下 (指解放)

機能 動物	先小文字”っ”
霊長類	左下 (指数字 4)

### 1.3. 片手文字

動かすことができる手が片方だけの場合はその手を左手、右手として機能させます。簡単のため、右手がないと仮定します。最初は左手を一次元座標的に左肩の辺りに配置し、あ行～わ行のどれかを選択します。

- ・左手：行選択

左手を指開放として右に移動させ、一次元座標的に首の辺りに配置します。

- ・左手：右移動

母音を選択します。

- ・左手：母音選択

濁音、半濁音、小文字に関する左手のシフトは以下のようになっています。

- ・濁音：指数字 1～5 で水平 (右または左) シフト
- ・半濁音：指数字 1～5 で上シフト
- ・小文字：指数字 1～5 で下シフト
- ・前小文字：左手を指開放として下シフト
- ・先小文字：左手を指数字 4 として下シフト
- ・前小文字：左手を右移動した後、左手を指開放として下シフト

その他は両手文字と同じです。

機能 動物	左数字	右数字	行シフト	濁音	半濁音	小文字	長音	セパレータ
霊長類	左手 (指数字:1,2,3,4,5)	左手	左上	左手水平	左上	左下	空間描画	左手右移動

機能 動物	は行	疑問符	前小文字
霊長類	左手指解放	左上 (半グー)	左下 (指解放)

### 1.4. 顔文字

動かすことができる手がない場合は顔のパーツを利用します。(1) は舌だけを用い、(2) は舌と目を併用します。

- ・あ行～な行 (1)：舌先左、舌先右、舌先中、舌先上、舌先下
- ・あ行～な行 (2)：舌先左、舌先右、舌先中、左目閉、右目閉

- ・は行～わ行 (1) : 舌尖左、舌尖右、舌尖中、舌尖上、舌尖下で首の左回転
- ・は行～わ行 (2) : 舌尖左、舌尖右、舌尖中、左目閉、右目閉で首の左回転
- ・わ行の要素 : わ、を、ん、 、?
- ・母音 (1) : 舌尖左、舌尖右、舌尖中、舌尖上、舌尖下で首の右回転
- ・母音 (2) : 舌尖左、舌尖右、舌尖中、左目閉、右目閉で首の右回転
- ・濁音 : 口をあの形で開けての首の右回転
- ・半濁音 : 上を向く
- ・小文字 : 下を向く
- ・ー : 口をあの形で開けて両目を閉じる
- ・? : 口をあの形で開けて右目を閉じる
- ・一文字後退削除 : 口を閉じ、両目を開けて首を左右に振る

濁音、半濁音、小文字はすべて前修飾動作です。

機能 動物	顔数字	母音	行シフト				
霊長類	舌尖:左, 右, 中, 上, 下 舌尖:左, 右, 中; 左目閉, 右目閉	首の右回転	首の左回転				

機能 動物			濁音	半濁音	小文字	長音
霊長類			口をあの形で開けての首の右回転	上を向く	下を向く	口をあの形で開けて両目を閉じる

### 1.5. 目文字

手、顔を用いて文字表現をすることができない場合は眼球の回転、瞼の開閉を利用します。行を表すパートと母音を表すパートには以下のような名称を与えます。

- ・あ行～な行、は行～わ行 : 第一バイト
- ・母音 (a, i, u, e, o) : 第二バイト
- ・あ行～な行 : 左、右、中、上、下
- ・は行～わ行 (1) : 左閉開の後、左、右、中、上、下
- ・は行～わ行 (2) : 左閉で左、右、中、上、下
- ・わ行の要素 : わ、を、ん、 、?
- ・第一バイト前のセパレーター : 左ダブルクリック
- ・第二バイト前のセパレーター : 右ダブルクリック
- ・母音 (1) : 右閉開の後、左、右、中、上、下
- ・母音 (2) : 右閉で左、右、中、上、下
- ・濁音 : 左閉で右半目
- ・半濁音 : 右閉で左半目
- ・小文字 : 両半目
- ・ー : やの次
- ・一文字後退削除 : ゆの次

濁音、半濁音、小文字はすべて前修飾動作です。

機能 動物	目数字	母音	行シフト			長音	セパレータ
霊長類	眼球:左, 右, 中, 上, 下	右閉	左閉			やの次	左ダブルクリック, 右ダブルクリック

機能 動物			濁音	半濁音	小文字		
霊長類			左閉右半目	右閉左半目	両半目		

## 2. 他の動物

他の動物では霊長類の手に対応する体の部位があれば学習は進歩的に継続するでしょう。足が四本ある犬、馬では多少バランスが乱れるものの1本を文字表現に使えます。象は鼻を霊長類の手のように自由自在に動かすことができます。鼻は一本であり、スピードは期待できませんが、鼻の先端の可動域が大きく、確実性が見込めます。

・犬、馬

機能 動物	左体数字	右体数字	行シフト	濁音	半濁音	小文字	長音
犬, 馬	左足 (左右:1,2; 前左中右:3,4,5)	右足 (同左)	顔左; 尾左	顔右	顔上	顔下	口開

・象

機能 動物	体数字	行シフト	濁音	半濁音	小文字	長音	セパレータ
象	鼻 (前下左右:1,2; 前上左中右:3,4,5)	左耳	鼻右奥	鼻頭近	鼻奥	口開	尾 (左, 右)

・海豚、海驢

機能 動物	体数字	行シフト	濁音	半濁音	小文字	長音	セパレータ
海豚, 海驢	1:舌先左,2:舌先右,3:舌先左→舌先右 4:左目閉,5:右目閉	顔左	顔右	顔上	顔下	口開	前びれ (左, 右)

・鳥

機能 動物	体数字	行シフト	濁音	半濁音	小文字	長音	セパレータ
鳥	左足 (左右前:1,2,3) 右足 (左右:4,5)	顔左	顔右	顔上	顔下	口開	翼 (左, 右)

3. 動物の言語教育

体話は動物が特別に優秀で、管理者が複数いる場合を除いて学習が滞る可能性が生じます。体数字は虚空、地面をポイントする場合はその位置の認識には不確実性が伴います。顔の変化は誤認識する可能性があります。尾の変化は見落とす可能性があります。そこで学習速度は遅くても教えやすい、わかりやすい手法を取り入れます。象やチンパンジーでは図1のようなデザインのシンボルの集まりを用いることにします。

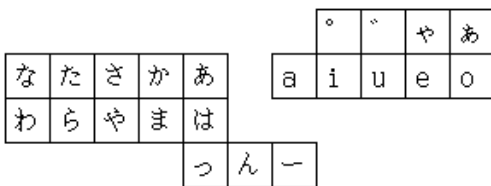


図 1

これはキーボードのキーの配置図であり、これを実画したものを画鍵またはグラフィックキーボードと称します。管理者は掲示し読み上げた結合かな文字に対応するタッチすべきシンボルを教え、タッチに誤りがあれば再タッチを促します。これを大盤指導と称します。以下のデバイスを付加すれば前回のキーボードに近いものになります。

- ・タッチセンサー
- ・ビープ
- ・LED(二色または一つのシンボルに二個)
- ・PC, Speaker

おやつは初めはシンボルの組み合わせ、時間的配置を理解してもらうために用います。ただし、動物は”待て”を遵守できなければなりません。習熟したら学習の終了後に与えます。

動物が最初に覚える言葉はおやつとします。おやつは特定の食料の名称ではなく食事の形態の一つであり、またその時に食べる食料の総称です。50音はおやつ要素の名称を覚えることで修得してもらいます。たとえば



- ・あんず
- ・かき
- ・すいか
- ・トマト
- ・なし
- ・バナナ
- ・もも
- ・ゆりね
- ・りんご
- ・わかめ

すべての50音を修得したら小文字を修得してもらいます。

- ・ちしや
- ・きゅうり
- ・みょうが
- ・ピッツァ
- ・キウイ
- ・トゥラウト
- ・チェリー
- ・ウォルナット

ただし、その動物に関する禁食食料は除外します。あ行の小文字は外来語が中心ですから文字数が多くなりがちです。短い言葉から始めて習熟したら長い言葉に移ると良いでしょう。

おやつ順序の例を以下に示します。

- ・もも、かき、なし、すもも、トマト、ゆりね、あんず、いちご、バナナ、パイナップル、ブルーベリー

ももの場合の手順は以下の通りです。

(1) ももの写真とかな文字”もも”が記されたボードを掲示し、”もも”と音読する

(2) かな行の”ま”にもものピースをあてがう

(3) ”ま”へのタッチを促す

(4) タッチしたらピースを与える

(5) 母音の’o’にもものピースをあてがう

(6) ’o’へのタッチを促す

(7) タッチしたら”も”と音読し、ピースを与える

(8) かな行の”ま”にもものピースをあてがう

(9) ”ま”へのタッチを促す

(10) タッチしたらピースを与える

(11) 母音の’o’にもものピースをあてがう

(12) ’o’へのタッチを促す

(13) タッチしたら”も”と音読し、ピースを与える

このピースフィードは普段のおやつにおいて同様のルーティーンを行うと難なくクリアできます。

かきの場合の手順は以下の通りです。

(1) かきの写真とかな文字”かき”が記されたボードを掲示し、”かき”と音読する

(2) かな行の”か”にかきのピースをあてがう

(3) ”か”へのタッチを促す

(4) タッチしたらピースを与える

(5) 母音の’a’にかきのピースをあてがう

- (6) 'a' へのタッチを促す
- (7) タッチしたら”か”と音読し、ピースを与える
- (8) かな行の”か”にかきのピースをあてがう
- (9) ”か”へのタッチを促す
- (10) タッチしたらピースを与える
- (11) 母音の'i'にかきのピースをあてがう
- (12) 'i' へのタッチを促す
- (13) タッチしたら”き”と音読し、ピースを与える

濁点 (゜)、半濁点 (゜゛)、小文字”っ”、”ん”、長音 (ー) が伴う場合はこれらのピースフィールドは母音のピースフィールドの前に行います。や行とあ行の小文字が伴う場合は母音を連続させます。たとえば”きゅ”では

- (2) かな行の”か”にピースをあてがう
- (3) ”か”へのタッチを促す
- (4) タッチしたらピースを与える
- (5) 小文字の”ゃ”にピースをあてがう
- (6) ”ゃ”へのタッチを促す
- (7) タッチしたらピースを与える
- (8) 母音の'i'にピースをあてがう
- (9) 'i' へのタッチを促す
- (10) タッチしたらピースを与える
- (11) 母音の'u'にピースをあてがう
- (12) 'u' へのタッチを促す
- (13) タッチしたら”きゅ”と音読し、ピースを与える

管理者のリードがなくなると動物は小文字の付加における母音の連続タッチができないかもしれません。その場合は大文字で代用してもかまいません。たとえば

・きゅうり

以上のようにして基礎を築いたら、現実的な名称の学習に移ります。

・あなた、わたし

”あなた”は動物から見た管理者です。管理者は自分を指差しながら”あなた”を音声と板書で教えます。”わたし”は動物ですから、その場で動物の写真を取り、動物を指差しながら”わたし”を音声と板書、写真で教えます。写真はなくても良いのですが、保険の意味で付加します。ただし、動物は”あなた”、”わたし”を各々人間達、動物達と誤解する可能性があります。

時間に関しては、まず夜と昼を教えます。

・よる、ひる

夜と昼はその写真で十分でしょう。

夜と昼を昨日、今日、明日の分組み合わせた時間軸を作り、今日には管理者と動物の写真を、昨日には昨日の証拠写真を添えます。時間軸の座標名は”じかん”とします。

・きのう、きょう、あした、じかん

これらと似たような言葉も教えます。

・まえ、いま、あと

この場合は別の時間軸を用います。一点を今とし、これに管理者と動物の写真を添えます。前と後は各々<、>に対応した半直線とし

ます。これらには異なる色または色調を付与します。

#### 4. 吃音治療

体話を吃音治療に応用してみます。特定の言葉に関して発話が不完全な場合を考えます。

- ・伸発：わーたしは
- ・連発：わわわたしは
- ・難発：... わたしは

治療発話において、クライアントは危機を感じたら直ちに体話または人工音声によるカバリングに移ります。

- ・体話 (十指表現)
- ・人工音声 (add10)

たとえば難発。”，”までが体話または人工音声です。

- ・わ，たしは
- ・わたしは，

筆者は難発の詳細を知らないのでカバリングが4文字分必要である後者の場合を考えます。最初のカバリングではクライアントは4文字すべてで閉口します。

- ・；；わたしは，

その後閉口をボーカルアクションに置き換えます。

- ・；わ；たしは，
- ・；わた；しは，
- ・；わたし；は，
- ・；わたしは；，

’；’がボーカルアクションのセクションの端点です。

逆順では

- ・わたしは；；，
- ・わたし；は；，
- ・わた；しは；，
- ・わ；たしは；，
- ・；わたしは；，

ボーカルアクションの種類は口パク、口腔発声、声帯発声です。体話と人工音声は一種のペースメーカー、デュエットパートナーと解釈することができます。

カバリングを行う前に addressing ten fingers の reading を口パク、口腔発声、声帯発声に関して十分行っておきます。

- ・addressing ten fingers：体話、人工音声 (50音+濁音+半濁音+ん+小文字)
- ・reading：口パク、口腔発声、声帯発声

”ん”と小文字の付加は当該ケースのかな結合だけでかまいません。危機的状況での”わ”をこの reading に条件反射的に完全な形で置き換えることができれば、クライアントは一般の社会生活においてはしばらくは体話を伴いながら発話することになるでしょう。成功の確率は伸発、連発 < 難発と見えています。

#### 5. 補遺

106 キーボードの無変換キーとひらがな・カタカナキーのベストポジションは以下の通りです。

- ・無変換キー：F と半キーだけバックスラッシュ型にオーバーラップ
- ・ひらがな・カタカナキー：L と半キーだけスラッシュ型にオーバーラップ

前回のスペックを表にすると以下ようになります。

機能	な行	た行	さ行	か行	あ行	a	i	u	e	o
Keyboard	な行	た行	さ行	か行	あ行	a	i	u	e	o
106	A	S	D	F	C	.	L	;	:	]

機能	っ	ー	ん	行シフト	半濁音	濁音	小文字や行	小文字あ行
Keyboard	っ	ー	ん	行シフト	半濁音	濁音	小文字や行	小文字あ行
106	無変換	Space	変換	ひらがな・カタカナ	O	P	@	[

キー割り当てを柔軟にすれば 106 キーボード以外のキーボードを使用することができます。親指シフトキーボードは親指キーの keysym(XK\_~) が存在することが条件です。

機能	な行	た行	さ行	か行	あ行	a	i	u	e	o
Keyboard	な行	た行	さ行	か行	あ行	a	i	u	e	o
富士通親指シフト	A	S	D	F	C	M	J	K	L	;

機能	っ	ー	ん	行シフト	半濁音	濁音	小文字や行	小文字あ行
Keyboard	っ	ー	ん	行シフト	半濁音	濁音	小文字や行	小文字あ行
富士通親指シフト	無変換	親指左	変換	親指右	U	I	O	P

もし PC のキーボードの知識があれば 106 キーボードを親指シフトキーボード風に改造できます。以下はその対応関係です。

- ・親指左  $\iff$  Space
- ・親指右  $\iff$  右 Ctrl

```
no.25::0 /* ideal */
old:else if(!BPideal && GKS(R11)<0) shift=1;
new:else if(!BPideal && GKS(XK_Ctrl_R)<0) shift=1;
```

ソースコードの 101 キーボードに関する修正箇所は get\_type() における R6、R11 と upper タイプの L11、L6、R4、R5、R10 です。

```
/* get_type(), else{}, left:106, right:101 */
・ R6=XK_Henkan  $\implies$  R6='m'
・ R11=XK_Hiragana_Katakana  $\implies$  R11=';'

/* get_type(), upper type, left:106, right:101 */
・ L11=XK_Muhenkan  $\implies$  L11='c'
・ L6=' '  $\implies$  L6='v'
・ R4='@'  $\implies$  R4='['
・ R5='['  $\implies$  R5=']'
・ R10='^'  $\implies$  R10='='
```

101 キーボードは upper タイプとして使用するので実行ファイル名の先頭に@をつけます。

- ・@add10

機能	な行	た行	さ行	か行	あ行	a	i	u	e	o
Keyboard	な行	た行	さ行	か行	あ行	a	i	u	e	o
101	Q	W	E	R	F	L	O	P	[	]

機能	っ	ん	ー	行シフト	半濁音	濁音	小文字や行	小文字あ行
Keyboard								
101	C	V	M	,	9	0	-	=

106 キーボードと solaris を組み合わせると Xlib では最下段の日本語キーが使えません。この場合は 101 キーボードに準じた修正を行います。

```
no.26::0    /* break */

/* get_type(), else{}, left:106, right:101 */
・ R6=XK_Henkan => R6='m'
・ R11=XK_Hiragana_Katakana => R11=','

/* get_type(), upper type, left:106, right:101 */
・ L11=XK_Muhenkan => L11='c'
・ L6=' ' => L6='v'

・ @add10
```

前回の手法も有効です。

```
no.25::1    /* ideal */
・ 配線の物理的スワップ
```

前回の図 1 のシステムにおいては、結合かな文字列の末尾に”っ”、”ん”がある場合は無限長音の付加を吟味しなければなりません。

- (1) ”あ” : 無限長音の付加あり
- (2) ”あー” : 無限長音の付加あり
- (3) ”あーっ”、”あーん” : 無限長音の付加あり
- (4) ”あっ”、”あん” : 無限長音の付加なし

(3) においてはシステムが”ー”を無限長音化します。したがって、結合かな文字列をその通りに放音するためには無限長音が break されたら”ーっ”、”ん”のサウンドを放音しなければなりません。

- ・ ”ーっ”のサウンドのシームレスな付加 : ”ー”の母音を用いる

フォントファイルの作成には DOS のビデオ BIOS1800h を利用します。

- ・ getfont.c : FreeDOS/V(fdos0138.img) 用。要 open-watcom-c-dos-1.9.exe
- ・ getfont.exe => DOSFONT1.BIN, DOSFONT2.BIN, SJS.BIN
- ・ add10.c : v1。要 nkf。set\_SB\_DB() において #if /\*UTF8\*/0 を #if UTF8 とする
- ・ add10 => sjis.bin, utf8.bin

\*\*\*\*\*

List 1:getfont.c

```
/* getfont.c */
/* MSDOS 3.1/V:device=a:\dos\himemx.exe */
/* _1800DX=0x0810 or 0x0813 */
/* COMMANDLINE:wcl gf.c -w -j -mh -bcl=dos */

/* 5 minutes hand sign */
/* usage:a situation in template(for example, a library) */
/* usage:dedication hand sign */
/*     part 1:solo(leading writer) */
/*     part 2:chokus(leading writer, two reflectors
/*         , the other writers)
/*     part 3:solo(leading writer)
/* usage:cheering squad members(
/*     leading writer, two reflectors, two interpreters)
/*     limb number on a hand
/*     1:pelvis
/*     2:stomach
/*     3:breastbone
/*     4:forward
/*     5:up
/*     shift on a hand:right or left(right interpreter only)
/*     shift on a hand:up
/*     shift on a hand:down
/* usage:a proper noun, a dialect, a vogue word
/* how to get the statistical data
/* duo or quartet
/* manuscript paper:a rotation by 90 degrees required
/* for(j=0;j<8;j++){
/* writer:{the original}_{j} into a line made up of squares
/* (a space is needed for near press, far press, 4-press)
/* writer:m_{all_j}
/* writer:write
/* reader:begin
/* reader:kana, kana+press(down arrow)
/*     , near press(|), far press(| with "hane"), 4-press(4)
/*     , backspace(left arrow)
/*     into the lower line made up of squares
/* down arrow:under the kana
/* 3 presses:down extension into the space field recommended
/* reader:end
/* reader:(delta_t)_{j}=end-begin
/* reader:n_{correct_j}
/*     }
/* each: $\sum (\delta_t)_{j} \equiv A, \sum n_{correct_j} \equiv B, \sum m_{all_j} \equiv C$ 
/* each:B/A, B/C
/* お、を:0.5
/* は、わ:0.5
/* standard writing rate:1W
/* the others:male-female-neuter, n-games
```

```

#include <stdio.h>
#include <stdlib.h>
#include <dos.h>
#include <jctype.h>

#define READ 0
#define WRITE 1

/* XMS */
#define Zbuf_X array_          /* array:EMS, array_:XMS */
#define _BYTES 1              /* 1 or 8 */
#define KBYTES (8*6*1280)     /* KBYTES*1024L bytes(ex. 1280x1024L) */

/* XMS */
unsigned xseg,xoff,handle_;
unsigned long XMS_CALL;
/* XMS, _BYTES=1,2,4,8 */
#if _BYTES==1
unsigned char far *buf;
unsigned char buff[1];
#elif _BYTES==2
unsigned int far *buf;
unsigned int buff[1];
#elif _BYTES==4
unsigned long far *buf;
unsigned long buff[1];
#else
dbl far *buf;
dbl buff[1];
#endif

typedef struct {
unsigned long Length; /* ]oCg */
unsigned int SourceHandle; /* ]nh */
unsigned long SourceOffset; /* ]ItZbg */
unsigned int DestHandle; /* ]nh */
unsigned long DestOffset; /* ]ItZbg */
} param;
param far *p;
param prm;

unsigned int _1800DX;
unsigned long base_,base[2];

int open_array_(void);
unsigned char array_(char,int,int,unsigned char,char,char);
void close_array_(void);
unsigned char work_p(char,int,int,unsigned char,char);
unsigned char work_Rect(char,int,int,unsigned char,char);

int main(int argc,char **argv)
{
unsigned char buf[3];
FILE *fp;

```

```

if(argc==1) _1800DX=0x0810;
else{
    if(!strcmp(argv[1],"0x0810)){
        _1800DX=0x0810;
    }
    else if(!strcmp(argv[1],"0x0813)){
        _1800DX=0x0813;
    }
    else{
        printf("bad value\n");
        return 1;
    }
}

if(open_array_()==1) return 1;
base[0]=0;
base[1]=1L*16*256;

set_SB_DB();

close_array_();

fp=fopen("SJS.BIN","rb");
fseek(fp,23*2L,0);
fread(buf,1,2,fp);
buf[2]='\0';
fclose(fp);

if(!strcmp(buf,"W"))
printf("BIOS call succeeded\n");
else
printf("BIOS call failed\n");

return 0;
}/** main **/

int set_SB_DB(void)
{
    unsigned char xs,ys,jis1,jis2,jis[2];
    unsigned char supp,pxl,m8,n8;
    unsigned short val[2];
    int j,i,m,n;                /* for(j,i,m,n) => int */
    unsigned int pcdx;
    FILE *fp_sjs2,*fp_fnt1,*fp_fnt2;

    typedef struct{
        unsigned char str[72];
    } tsMIB;
    tsMIB MIB;
    tsMIB /*far*/ *mib;

    if(0){
}/**if()**/

```



```

else if(1){
fp_fnt1=fopen("dosfont1.bin","w+b");

mib=&MIB;
val[0]=FP_SEG(mib);
val[1]=FP_OFF(mib);

#if 1
pcdx=_1800DX;
for(n=0x20;n<=0x7e;n++){          /* begin:0x20 */
n8=n;
_asm{
push es
push si
mov ah,18h
mov al,00h
mov bh,0
mov bl,0
mov ch,0x00
mov cl,n8 ;41h:A,5fh:_
mov dx,pcdx
;mov dx,0x0810 ;HANKAKU(pcdos,msdos)
;mov dx,0x0813 ;HANKAKU(freedos)
mov si,val[0]
mov es,si
lea si,[MIB]
int 10h
mov supp,al
pop si
pop es
}

if(0) printf(" supp:%d\n",supp);

/* to XMS */
if(!supp){
for(j=0;j<=15;j++){
/*printf(" %d",MIB.str[j]);*/
work_p(WRITE,0x00,n,MIB.str[j],j);

fprintf(fp_fnt1,"%c",MIB.str[j]);
}
}/**if(!supp)**/
else{
for(j=0;j<=15;j++) fprintf(fp_fnt1," ");
}/**else(!supp)**/
}/**for(n)**/
#endif

fclose(fp_fnt1);

fp_sjs2=fopen("sjs.bin","w+b");
fp_fnt2=fopen("dosfont2.bin","w+b");

/*mib=&MIB;

```

```

val[0]=FP_SEG(mib);
val[1]=FP_OFF(mib);*/

for(m=0;m<=0xff;m++){
for(n=0;n<=0xff;n++){
m8=m;
n8=n;
if(iskanji(m8) && iskanji2(n8)) ;
else{
for(j=0;j<=31;j++) fprintf(fp_fnt2," ");
goto end;
}
_asm{
push es
push si
mov ah,18h
mov al,00h
mov bh,0
mov bl,0
mov ch, m8
mov cl, n8 ;0x82a0:
mov dx,0x1010 ;ZENKAKU
mov si,val[0]
mov es,si
lea si,[MIB]
int 10h
mov supp,al
pop si
pop es
}

/*99*/
if(0) printf(" supp:%d\n",supp);

/* to XMS */
if(!supp){
for(j=0;j<=31;j++){
/*printf(" %d",MIB.str[j]);*/
work_Rect(WRITE,m,n,MIB.str[j],j);

fprintf(fp_fnt2,"%c",MIB.str[j]);
}

fprintf(fp_sjs2,"%c%c",m8,n8);
}/**if(!supp)**/
else{
for(j=0;j<=31;j++) fprintf(fp_fnt2," ");
}/**else(!supp)**/
end;;
}/**for(n)**/
}/**for(m)**/

fclose(fp_sjs2);
fclose(fp_fnt2);
}/**else**/

```

```

return 0;
}/** set_SB_DB **/

unsigned char array_(char flag,int nx,int ny,unsigned char val,char bytes,char dlt)
{
if(nx<0 || nx>0xff || ny<0 || ny>0xff) return 0; /* XRES0-1,YRES0-1 => 0xff */

if(flag==0){
prm.Length=/*bytes*/2;
prm.SourceHandle=handle_;
prm.SourceOffset=base_+(unsigned long)nx*256*bytes+ny*bytes+dlt;
prm.DestHandle=0;
prm.DestOffset=(unsigned long)buf;

copyxms(xseg,xoff);

val=buff[0];
}
else{
buff[0]=val;

prm.Length=/*bytes*/2;
prm.SourceHandle=0;
prm.SourceOffset=(unsigned long)buf;
prm.DestHandle=handle_;
prm.DestOffset=base_+(unsigned long)nx*256*bytes+ny*bytes+dlt;

copyxms(xseg,xoff);
}

return val;
}/** array_ **/

int open_array_(void)
{
/* XMS OK? */
if(initxms(KBYTES,&handle_)!=1) {printf(" ?\n");return 1;}
printf(" handle=%x\n",handle_);

p=&prm;
buf=buff;

xseg=FP_SEG(p);
xoff=FP_OFF(p);

return 0;
}/** open_array_ **/

void close_array_(void)
{
unsigned Handle,val;

```

```
val=closetxms(handle_,&Handle);
printf(" ax=%d\n",val);
printf(" freed_handle=%x\n",Handle);
}/** close_array_ **/
```

```
int initxms(unsigned kbytes,unsigned *Handle)
```

```
{
unsigned flag,hdl;

_asm{
push es

mov ax,4300h
int 2fh
cmp al,80h
jne end

mov ax,4310h
int 2fh
mov word ptr XMS_CALL,bx
mov word ptr XMS_CALL+2,es
clc

mov ah,09h
mov dx,kbytes ;kbytes
call [XMS_CALL]

mov hdl,dx ;handle
mov flag,ax
jmp end_

end:
mov flag,0h

end_:

pop es
}
```

```
*Handle=hdl;
```

```
return flag;
}/** initxms **/
```

```
int copyxms(unsigned xseg,unsigned xoff)
```

```
{
unsigned flag;

_asm{
push ds
push si
```

```
mov ds,xseg
mov si,xoff
mov ah,0bh
call [XMS_CALL]
```

```
mov flag,ax
```

```
pop si
pop ds
}
```

```
return flag;
}/** copyxms **/
```

```
int closexms(unsigned handle,unsigned *Handle)
{
unsigned flag,hdl;
```

```
_asm{
mov ah,0ah
mov dx,handle
call [XMS_CALL]
```

```
mov hdl,dx
mov flag,ax
}
```

```
*Handle=hdl;
```

```
return flag;
}/** closexms **/
```

```
unsigned char work_Rect(char RW,int x,int y,
unsigned char pcolor,char dlt)
```

```
{
unsigned char val;
```

```
base_=base[1];
```

```
if(!RW){ /* read */
val=Zbuf_X(READ,x,y,-1,32,dlt);
}
else{ /* write */
Zbuf_X(WRITE,x,y,pcolor,32,dlt);
}
```

```
return val;
}/** work_Rect **/
```

```
unsigned char work_p(char RW,int x,int y,
unsigned char pcolor,char dlt)
```

```
{
```

```
unsigned char val;
```

```
base_=base[0];
```

```
if(!RW){ /* read */  
val=Zbuf_X(READ,x,y,-1,16,dlt);  
}  
else{ /* write */  
Zbuf_X(WRITE,x,y,pcolor,16,dlt);  
}
```

```
return val;  
}/** work_p **/
```