

제 3 장 A/D 변환과 PC/AT

1. 목 적

실험과목을 진행하다 보면 대부분의 학생들이 디지털표시기를 좋아하는 경향이 있고, Sensor 부분보다는 Read-Out 장치에 매혹되는 경우가 많다. 그러나 자연의 대부분의 신호는 아날로그이며, 화려한 Read-Out 장치도 Sensor의 신호를 처리한 것에 불과하다. 따라서 Sensor에서 나오는 원래의 아날로그 신호가 잘못된 것이라면, 디지털로 아무리 잘 변환하여도 엉터리일 수 밖에 없다. 따라서 학생 여러분들은 좀 읽기가 불편하더라도 가급적이면 Sensor 근처에서 나오는 아날로그 신호를 읽어내려는 노력을 하는 것이 좋다.

그러면 Computer로 Control되는 A/D 변환은 왜 할까? 여러개의 『Black Box』를 통과한 신호는 더 믿을 수 없을텐데도 굳이 해야 하는 이유가 있다. 첫째, 정확한 시간제어를 위해 필요하다. 우리가 시계를 보며 잴 수 없을 만큼 짧은 시간간격, 예를들면 1msec 간격으로 데이터를 받아들여야 한다면 인간의 눈으로 읽지 못할 것이다. 또 동시에 몇개의 신호를 처리하여야 경우도 있다. 둘째, 데이터 양이 방대한 경우에도 Computer가 연결되지 않은 상태로 실험을 한 어떤 사람이, 데이터량이 많아 Roll로 된 데이터 용지를 어깨에 메고 와서 몇일 동안 컴퓨터에 입력시키는 것을 본적이 있다.

여기서는 Signal Conditioner를 통과한 양호한 신호를 A/D 변환하여 Computer에서 받아 들이게 하는 방법을 익히도록 한다.

2. 예습부분

- (1) Mechanical Measurements(4th Ed.), T. G. Beckwith & R. D. Marangoni
Chapter 8. (pp. 292-300)
- (2) PC/AT Disk Operating System
- (3) BASIC

3. 실험장치

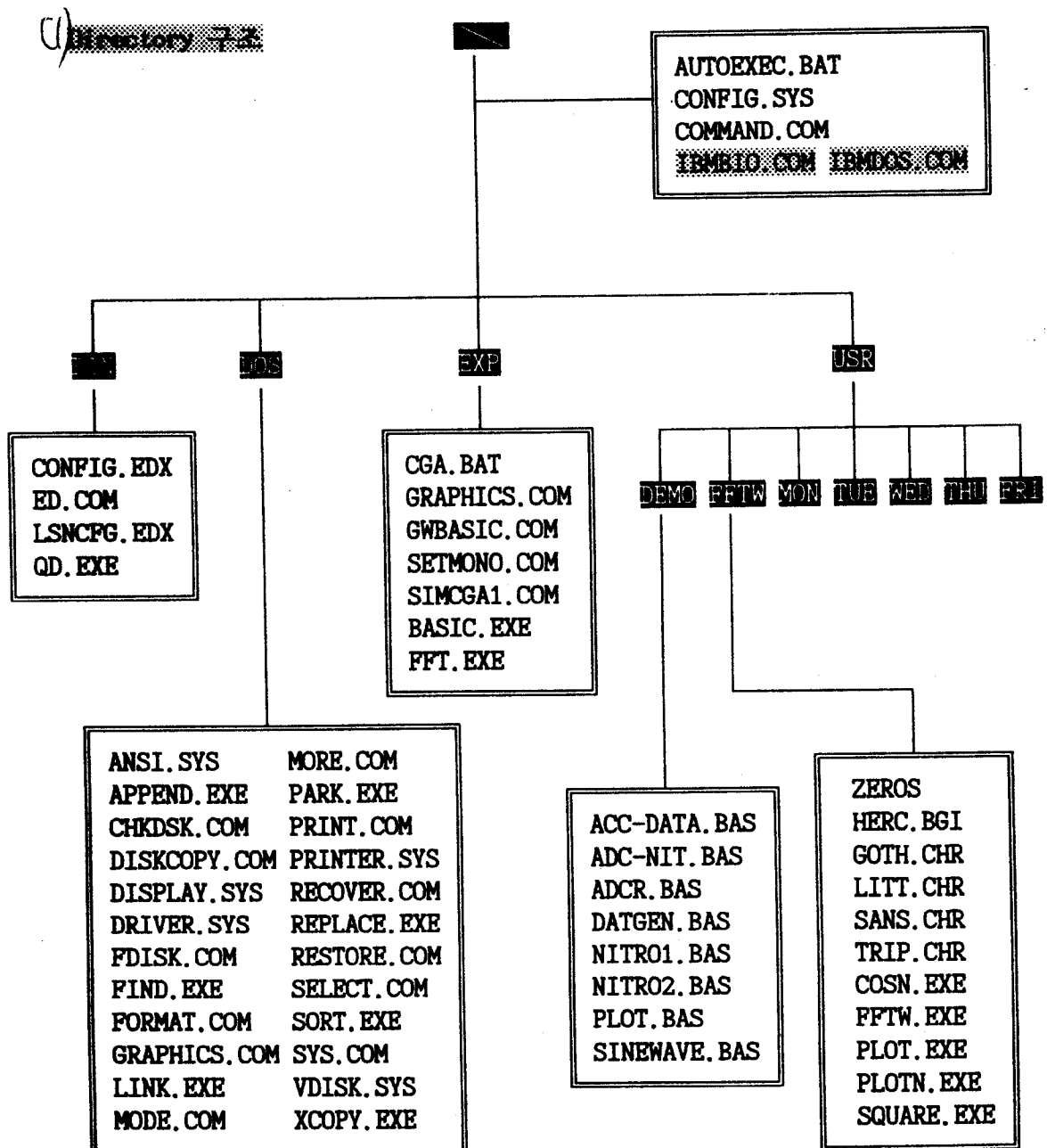
- (1) Oscilloscope, FG, DMM
- (2) Epson Printer

(3) A/D Converter Board(DT2814)

(4) 2DD/2HD 5.25" Diskette 각자 지참

4. 이 론

4.1 PC/AT MS-DOS 요약



(2) **Hardware PC/AT**

①

(1) System Units

- * CPU : Intel 80286 micro-processor
- * Hercules Monochrome Graphics Adaptor
- * RAM Size : 640KB
- * Parallel Port
- * RS-232C Serial Port
- * 80287 Co-Processor
- * DT-2814 A/D Converter Board

②

(2) Disk Drives

- * 1 Floppy Disk Drive (5.25")
- * 20MB Hard Disk

(3) **GETTING STARTED & ENDED**

① (1) Power ON (Wait about a minutes for memory check)

② (2) Boot DOS (MS DOS V3.3 or later)

* Warm boot : <Ctrl>-<Alt>-

* DOS files : IBMBIO (IO. SYS)

IBMDOS (MSDOS.SYS)

COMMAND.COM - Command processor

(4) **OPERATING SYSTEM : MS-DOS (or PC-DOS)**

①

(1) DOS device names (Hard disk) : C:

②

(2) DOS Prompter : A:\> , C:\>

③

(3) Device change : C:\> A: [return]

④

(4) DOS file organization : TREE structure

Directory

File

⑤

(5) Command type : Internal and external

⑥

(6) Filespec : combination of device name, path and file name

⑦

(7) Filename : 8 character + 3 character extension

⑧ Reserved file names

ASSIGN	DATE	GRAPHICS	RD
BASIC	DEBUG	LINK	RMDIR
BASICA	DEL	MKDIR	SET
BREAK	DIR	MODE	SORT
CD	DISKCOMP	MORE	TIME
CHDIR	DISKCOPY	PATH	TREE
CHKDSK	EDLIN	PRINT	TYPE
CLS	ERASE	PROMPT	VER
COMMAND	EXE2BIN	RECOVER	VERIFY
COMP	FDISK	RENAME	VOL
COPY	FIND	REN	
CTTY	FORMAT	RESTORE	

⑨ Reserved extension

.COM	.EXE	.BAT	.SYS
.BAS	.FOR	.LIB	.OVR
.BAK			

⑩ DOS File commands

* Global filenames (wild cards) : * and ?

MD <path> : make directory
 CD <path> : change directory
 RD <path> : remove directory
 DIR <path> : file and directory list
 COPY <source filespec> <target filespec>
 TYPE <filespec>
 REN <old filespec> <new filespec>
 DEL <filespec>
 CHKDSK
 VERIFY (ON|OFF)
 FORMAT A: /S /V
 DATE, TIME
 BASIC
 CLS
 DISKCOPY
 TYPE <filespec>

4.2 IBM-BASIC 요약

(1) Getting Started

Enter into BASIC mode : C:\> CGA
C:\> BASIC [enter]
Exit from BASIC : SYSTEM [enter]
BASIC Prompter : OK,
LOAD <filename>
SAVE <filename> [,A]

(2) Editor Commands

BEEP
LIST (F1)
NEW
RENUM
RUN
SYSTEM
TRON, TROFF

(3) Basic Syntax

CLS
READ, DATA, RESTORE
DIM
END, STOP
FOR ... NEXT
GOSUB ... RETURN
ON <number> GOSUB N1,N2,N3 ...
GOTO <number>
IF .. THEN.. ELSE
INKEY\$
INPUT
LOCATE <row>, <col>
PRINT
PRINT USING "###.###" ; variable
REM
SWAP

(4) Operators

^, -, +, *, /, \, =, <> or ><, <, >, <= or =<, >= or =>

NOT, AND, OR

String : A\$ = " My name is Hong "

(5) Graphics using CGA/BAT

PSET (x, y)

LINE (x1, y1)-(x2, y2), LINE -(x2,y2) LINE (x1,y1)-(x2,y2),,B

CIRCLE (X,y), r[,color,start,end,aspect ratio]

VIEW (ix0,iy0)-(ix1,iy1)

* Monitor resolution : 720x348

WINDOW (xmin,ymax)-(xmax,ymax)

(6) File Operations

FILES "A:

OPEN "filename" FOR

INPUT
OUTPUT
APPEND

 AS #number

PRINT #number, x, y, ...

INPUT #number, x, y, ...

CLOSE #number

Numeric Functions	
Function	Result
Arithmetic	
ABS(x)	Absolute vlaue of x
ATN(x)	Arctangent (in radians) of x
CDBL(x)	Converts x to a double precision number
CINT(x)	Converts x to an integer by rounding
COS(x)	Cosine of angle x, where x is in radians.
CSNG(x)	Converts x to a single precision number
EXP(x)	Raises e to the x power
FIX(x)	Truncates x to an integer
INT(x)	Largest integer less than or equal to x
LOG(x)	Natural logarithm of x
RND(x)	Random number
SGN(x)	Sign of x
SIN(x)	Sine of angle x, where x is in radians
SQR(x)	Square root of x
TAN(x)	Tangent of angle x, where x is in radians

Numeric Functions (Continued)	
Function	Result
String-Related	
ASC(x\$)	ASCII code for first character in x\$
CVI(x\$) CVS(x\$)	Convert x\$ to a number of integer single or double precision
CVD(x\$)	
INSTR(n,x\$,y\$)	Position of first occurrence of y\$ in x\$ starting at position n
LEN(x\$)	Length of x\$
VAL(x\$)	Numeric value of x\$
I/O and Miscellaneous	
CSRLIN	Returns the vertical line position of the cursor
EOF(f)	Indicates and end-of-file condition on file f.
ERL	Returns the line number where the last error occurred (<i>see</i> ERR)
ERR	Returns the error code number of the last error
FRE(x\$)	Returns the amount of free space in memory not currently in use by BASIC
INP(n)	Reads a byte from port n
LOC(f)	Returns the "location" of file f <ul style="list-style-type: none"> • next record number of random file • number of sectors read or written for sequential file • number of characters in communications input buffer
LOF(f)	Returns the length of file f <ul style="list-style-type: none"> • number of bytes (in multiples of 128) in sequential or random file • number of bytes free in communications input buffer
LPOS(n)	Returns the carriage positions of the printer
PEEK(n)	Reads the byte in memory location n
PEN(n)	Reads the light pen
PLAY(n)	Number of notes in Music Background buffer
PMAP(x,n)	Maps an expression to world or physical coordinates
POINT(n)	Value of current x or y graphics coordinate(LRP)
POINT(x,y)	Returns the color of point(x,y) (graphics mode)
POS(n)	Returns the cursor column position
SCREEN(row,col,z)	Returns the character or color at position (row,col)
STICK(n)	Returns the coordinates of a joystick
STRIG(n)	Returns the state of a joystick button
TIMER(n)	Number of seconds elapsed since midnight/system reset
USRn(x)	Calls machine language subroutine with argument x
VARPTR(variable)	Returns the address of the variable in memory
VARPTR(#f)	Returns the address of the file control block for file f

String Functions	
Function	Result
CHR\$(n)	Character with ASCII code n
LEFT\$(x\$,n)	Leftmost n characters of x\$
MID\$(x\$,n,m)	m characters from x\$ starting at position n
RIGHT\$(x\$,n)	Rightmost n characters of x\$
SPACE\$(n)	n spaces
STRING\$(n,x\$)	First character of x\$ repeated n times
STRING\$(n,m)	Character with ASCII value m, repeated n times
I/O and Miscellaneous	
DATE\$	Returns the system date
HEX\$(n)	Converts n to a hexadecimal string
INKEY\$	Reads a character from the keyboard
INPUT\$(n,#f)	Reads n characters from file f
MKI\$(x), MKS\$(x), MKD\$(x)	Converts x in indicated precision to proper length strength
OCT\$(n)	Converts n to an octal string
SPC(n)	Prints n spaces in a PRINT or LPRINT statement
STR\$(x)	Converts x to a string value
TAB(n)	Tabs to position n in a PRINT or LPRINT statement
TIME\$	Returns the system time
VARPTR\$(v)	Returns a three-byte string containing the type of variable and the address of the variable in memory

4.3 Analog-to-Digital Converter

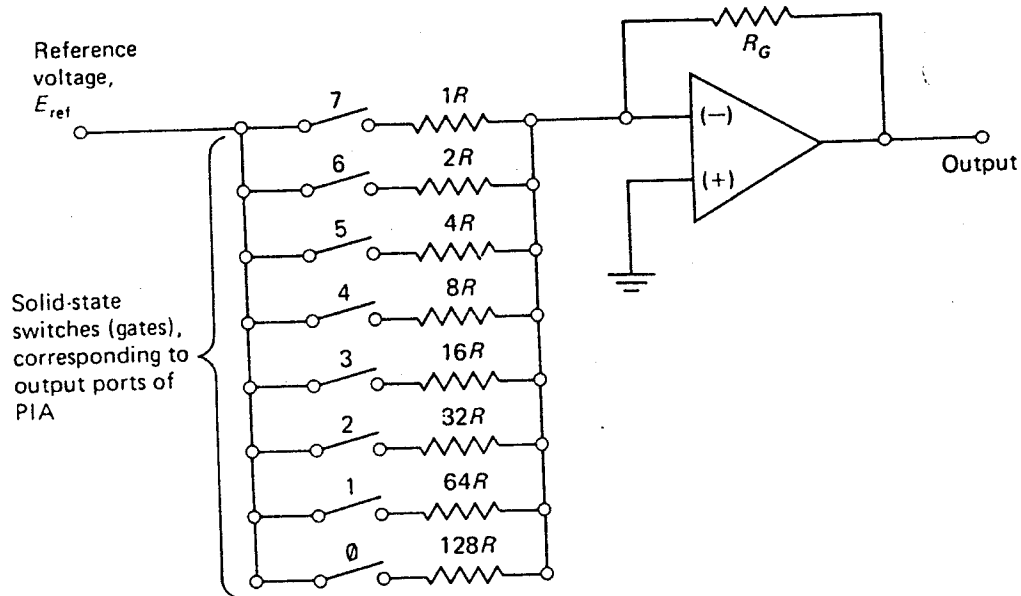


Fig.3.1 A simple DAC (digital-to-analog converter)

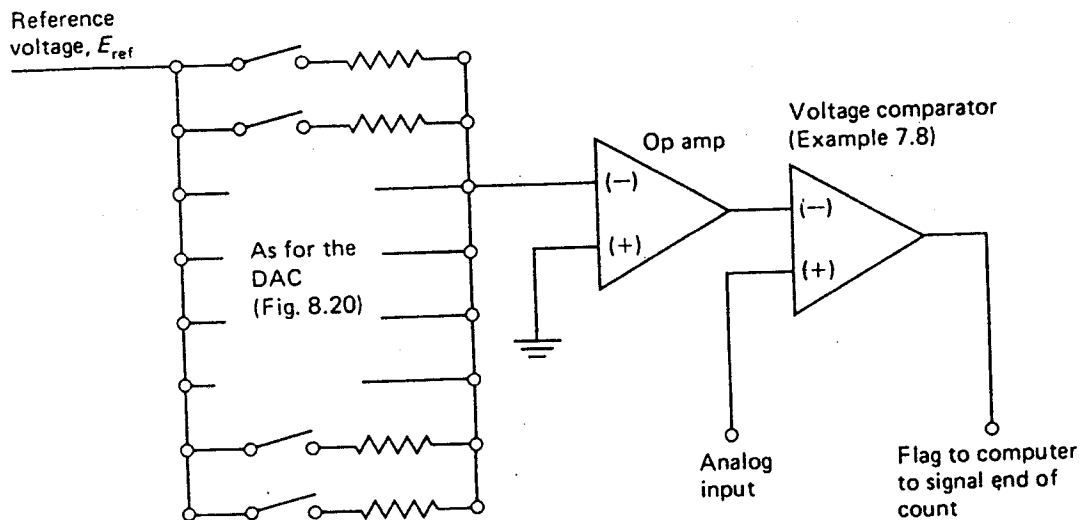


Fig.3.2 A simple ADC (analog-to-digital converter)

Fig.3.1, Fig.3.2는 Reference에 나오는 그림으로서 D/A Conversion 과 A/D Conversion의 원리를 간단히 표시한 것이다. Reference의 내용을 참고하여 잘 생각해 보라 다음은 Data Translation 회사에서 PC XT/AT용으로 개발한 A/D Converter Interface DT2814의 규격 및 사용법을 정리한 것이다.

(1) Features

- * DATA TRANSLATION
- * 16 Channels, 12 bit resolution
- * Input ranges 0~5 volts, ± 5 volts, ± 2.5 volts (Jumper selectable)
- * Maximum input without damage
Power ON : ± 30 V, Power OFF : ± 20 V
- * A/D conversion time : 25 μ sec.
- * Channel acquisition time : 15 μ sec.

(2) Resistor description

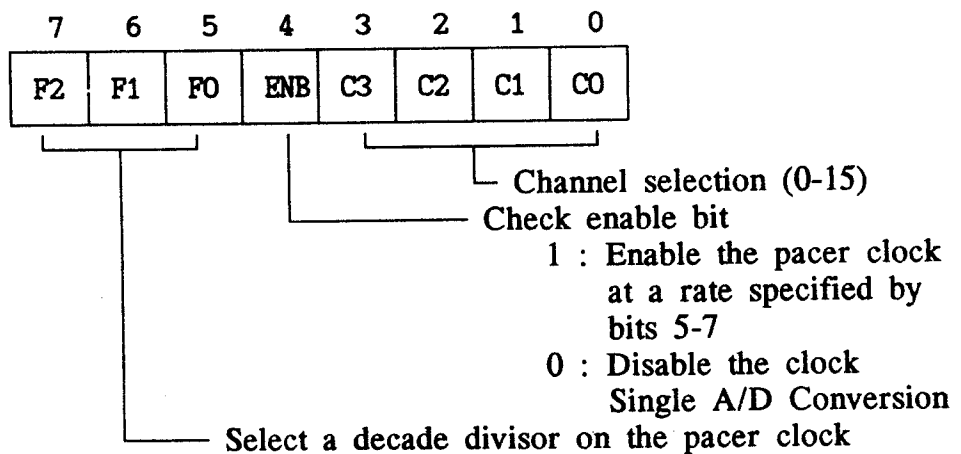
- * Location in I/O space : 200H - 3FEH

(3) Control/Status resistor (CSR)

- * Located at the base address
- * Control and monitor all activity
- * Write access and read access

(4) Write access

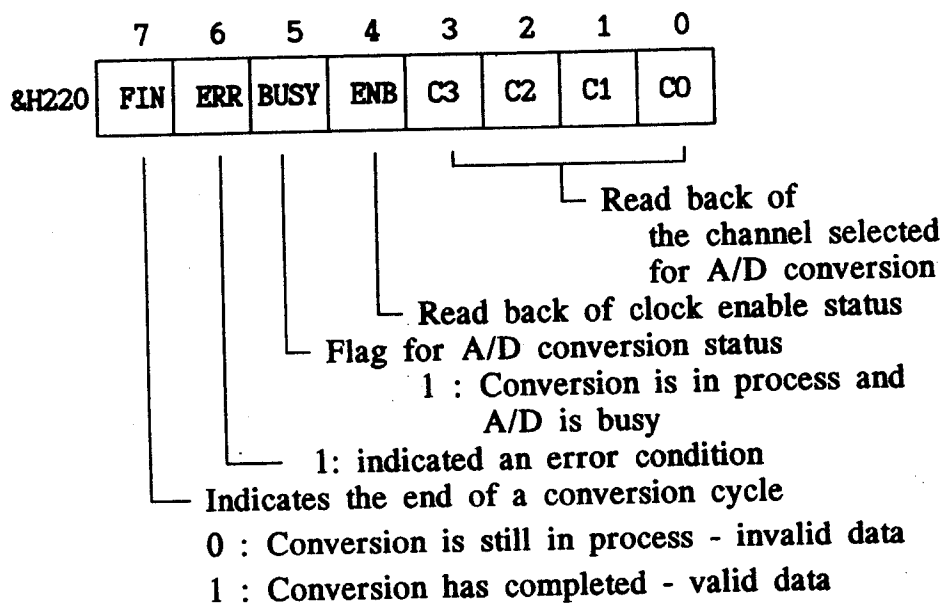
- * Select the A/D channel
- * The clock decade divisor
- * Enable or disable the pacer clock for A/D conversion
- * Bit description



(5) Read access

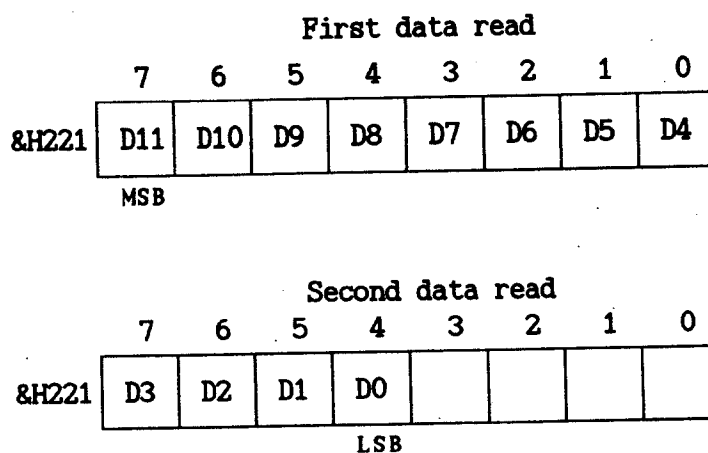
- * Shows the status of the channel of currently selected the clock enable status, the status of the A/D converter and the error status.

*** Bit description**



(6) A/D data resistor

- * Located at the base address + 1
- * Located in the order of high byte - low byte
- * All 12 bits are left justified
- * Bit definitions



*** Data format**

Offset binary for a bipolar range

Straight binary for a unipolar range

ADC-DATA.BAS

```
1000 CLS
1010 DIM V1(256),V2(256),V3(256)
1020 Y=15
1030 PRINT "ch ";Y
1040 FOR I=1 TO 256
1050 OUT &H220,Y
1060 A=INP(&H220)
1070 IF A>=128 THEN 1080 ELSE 1060
1080 V1(I)=INP(&H221)
1090 V2(I)=INP(&H221)
1100 NEXT I
1110 PRINT:PRINT:PRINT "***** completed!!"
1115 PRINT:PRINT :PRINT
1120 FOR I=1 TO 256
1130 VALUE=V1(I)*16+V2(I)/16
1140 VOLT=(VALUE-2048)/2048*5
1150 V3(I)=VOLT
1160 PRINT I,VALUE,VOLT
1170 NEXT I
1180 REM *****
2000 PRINT "type output file name"
2010 INPUT D$
2020 OPEN D$ FOR OUTPUT AS #1
2030 FOR I=1 TO 256
2040 PRINT #1,V3(I)
2050 NEXT I
2060 END
```

실 험 3 : PC/AT의 사용법

1. MS-DOS

책 참고

2. BASIC Programing Language

책 참고

3. 간단한 IBM PC-XT 사용법

(1) BOOTING

- Power switch를 on 시킨다.

(2) REBOOTING

- [Ctrl]+[Alt]+[Del] key를 동시에 누른다.

(3) Diskette에 들어 있는 file 확인 : C:\> dir a:

(4) Disk drive C:\>에서 A:\>로의 이동 : C:\> A:

(5) Basic 구동 : C:\> BASIC

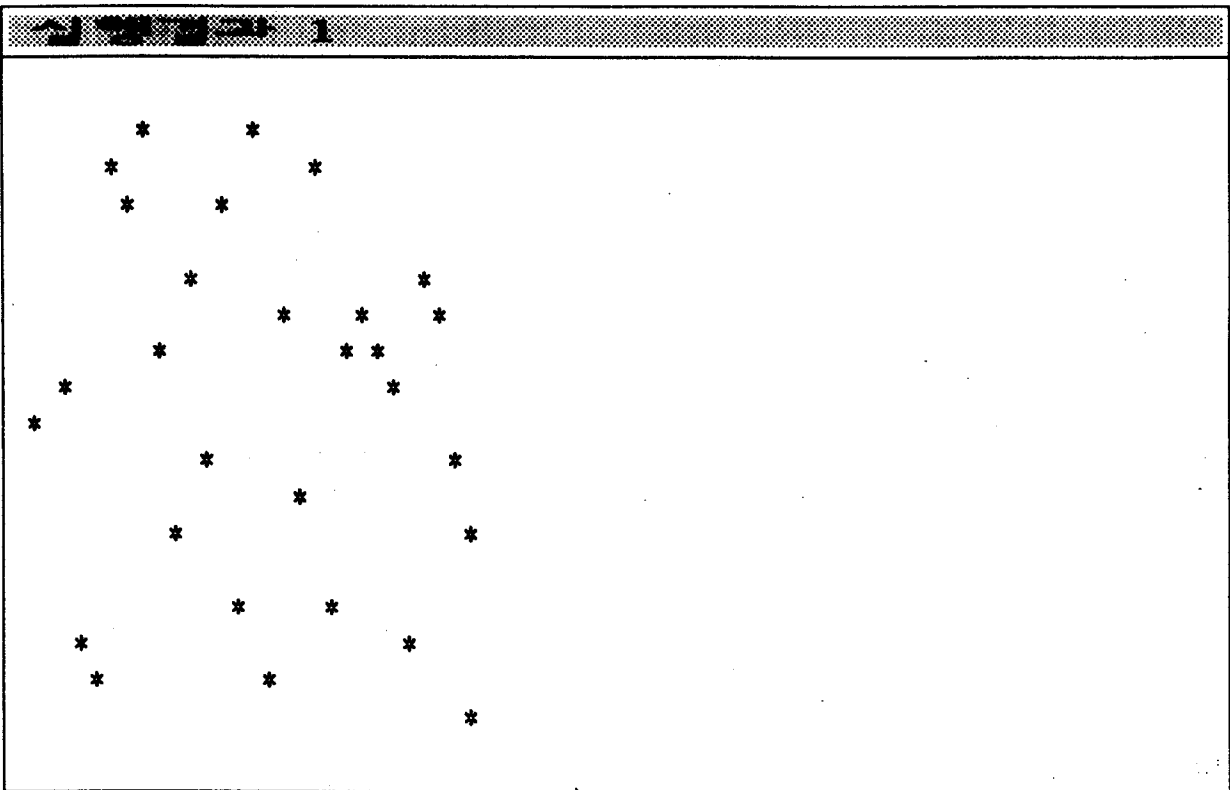
(6) Basic에서 System으로 SYSTEM

4. 프로그램 연습

다음과 같은 결과를 얻기 위한 Program을 각자 Programing 해 본다.

이때 $y = A \sin (Bx+C)$ 의 형태에서 A, B, C를 받아들여서 그때의 sine wave가 나타날 수 있게 Programming 한다.

```
LIST
10 CLS
20 FOR N=1 TO 30
30 Y = INT ( 24*RND(1) + 1 )
40 x = INT ( 55*RND(1) + 1 )
50 LOCATE Y,X
60 PRINT "*"
70 NEXT N
80 END
OK
```



실험 결과 2

```

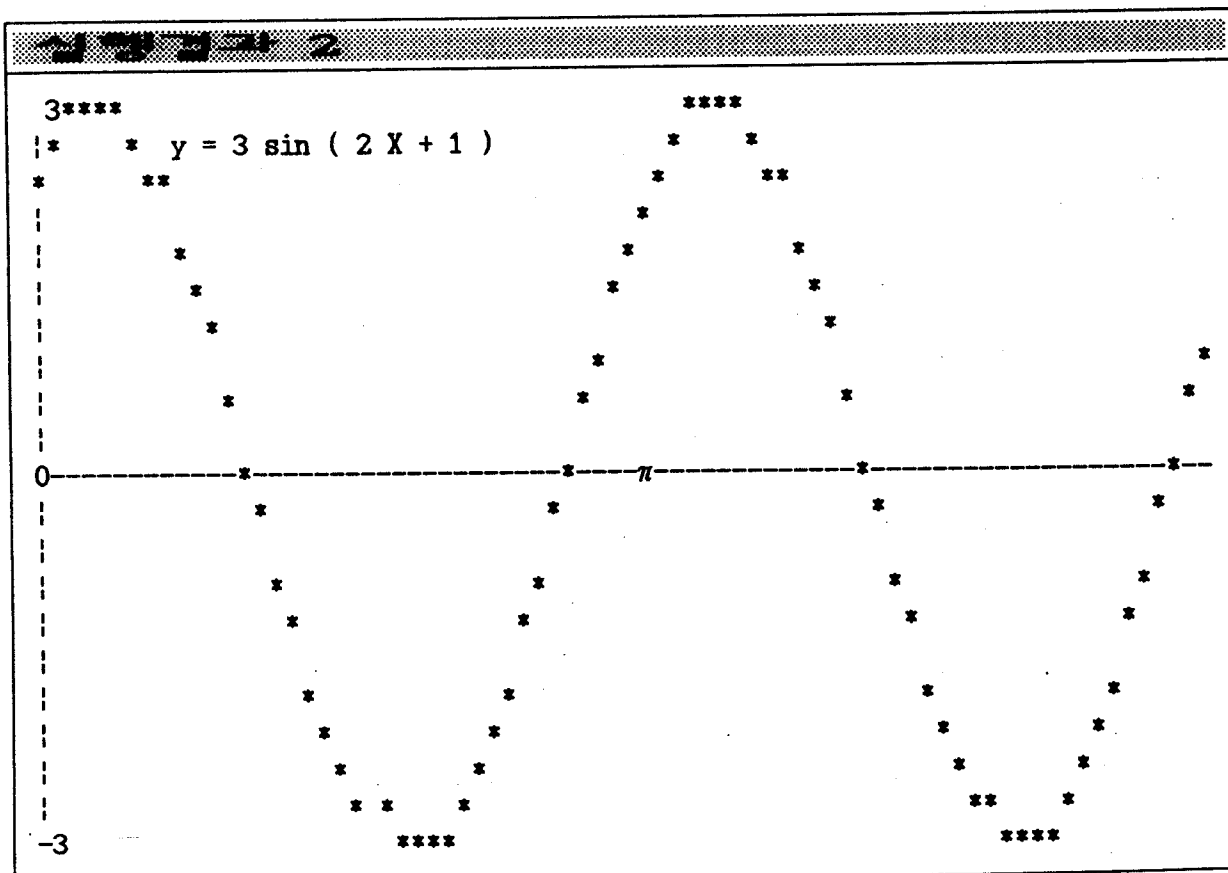
700 REM **** This is a program of a graph of  $y = a * \sin(b*x+c)$  ****
800 '
900 ' *****      input program      *****
1000 CLS
1020 PRINT " ***** graph of  $y=a*\sin(b*x+c)$  *****"
1100 LOCATE 6,10 : INPUT "amplitude = " ; A
1200 LOCATE 8,10 : INPUT "frequency = " ; B
1300 LOCATE 10,10 : INPUT "phase(radian) = " ; C
1400 CLS
1500 BEEP:BEEP:BEEP:BEEP
1510 ' *****      program of graphing axes ,x, y      *****
1600 FOR I=1 TO 21
1700 LOCATE I,1 : PRINT "|"
1800 NEXT I
1900 FOR J=1 TO 80
2000 LOCATE 11, J :PRINT "-"
2100 NEXT J
2110 ' *****      program of graphing  $y=a*\sin(b*x+c)$       *****
2200 FOR K=1 TO 79

```

```

2300 PI=3.141592
2400 X1=(K-1)*PI/39
2500 Y=A*SIN(B*X1+C)
2600 Y1=INT(11-10/A*Y+.5)
2700 LOCATE Y1, K : PRINT "*"
2800 NEXT K
2900 LOCATE 1,1 :PRINT A
3000 LOCATE 11,1 :PRINT "0"
3100 LOCATE 11,40 :PRINT CHR$(227)
3200 LOCATE 11,78 :PRINT "2" ;CHR$(227)
3300 LOCATE 21,1 :PRINT -A
3400 LOCATE 2,10 :PRINT "y = "; A; "sin (" ; B; "x + " ; C ; ")"
3500 GOTO 3500

```



```

100 CLS
200 BEEP
300 C$ = "-----"
400 A$ = "*"
500 B$ = "|"
600 PRINT "-1"; TAB(34); "0"; TAB(68); "1"
700 PRINT C$;A$;C$
800 FOR X=18 TO 360 STEP 18
900 X1=X/180*3.141592
1000 Y=SIN (X1)
1100 Y1=35+INT(Y*34+.5)
1200 IF Y1>35 THEN PRINT TAB(35) ;B$ ; TAB(Y1) ;A$
1300 IF Y1=35 THEN PRINT TAB(35) ;A$
1400 IF Y1<35 THEN PRINT TAB(Y1) ;A$ ; TAB(35) ;B$
1500 NEXT X
1600 END
OK

```

