PLEASE

TABLE OF CONTENTS

INSTRUCTIONS

Introduction to PLEASE

I have been working with minicomputers and microcomputers for nearly eight years. During this time I have seen some pretty horrible examples of how NOT to design small computer software packages. Often, concepts and techniques which might be appropriate to a large computer system are thoughtlessly applied to a small system, with predictably bad results. I have developed a number of programming philosophies and techniques which are designed specifically for the types of small machines that are becoming available to the computer hobbyist and small businessman.

PLEASE is a very small package which embodies several of the more important concepts. It has a tiny monitor which takes care of the basic I/O functions, timing functions, and scheduling. It has a set of functions to perform the major tasks. These functions are combined through the use of a simple interpreter which calls up the functions and passes the necessary parameters to them. Programs are written as a series of functions. Since each function does a major piece of work, a program may consist of only a few functions and can be readily "assembled" by hand. For example, the PLEASE Command Decoder which permits the user to select which game or demo to run from the keyboard is four functions, encoded in a total of 16 bytes.

I am very impressed with the capabilities of the KIM-1 Microcomputer System. Aside from the excruciatingly slow cassette dump/load, it is remarkable well thought out. However, I must admit that after getting my own KIM-1 and doing the single example in the manual I was more than a bit frustrated that there wasn't anything else to do without a great deal of effort on my part. I could only impress my wife and friends slightly by adding 2 + 3 and getting 5! I decided, therefore, to create a small games and demos package that could run on the basic KIM-1, without any additional peripherals such as teletype or special display devices. The result is PLEASE.

PLEASE has been developed entirely on a basic KIM-1. All of the assembly has been done by hand. All of the object code has been entered through the keyboard. Debugging has been via the KIM-1 Monitor. The resulting package demonstrates what can be done on a basic system. It also does suffer in some respects though. Certain changes I would have liked to make became just too much effort via hand assembly.

I welcome any suggestions, comments, or whatever. I have had fun developing PLEASE and hope you will enjoy playing with it.

Robert M. Tripp

Prices for the PLEASE Package are:

OBJECT CASSETTE TAPE and Operating Instructions:	$6.00
SOURCE LISTINGS and Instructions for Writing Code:	$6.00
Complete PLEASE Package Including Everything:	$10.00

These may be ordered from the above address. For information: 617/256-364-

General Operating Instructions

Getting Started.

The PLEASE Package is divided into a number of components which are stored separately on the cassette tape.

Conceptually, the PLEASE Package is divided into four major components:

1. KIM-1 Monitor
2. PLEASE Executive
3. PLEASE Functions
4. PLEASE Programs

In order to run a PLEASE program, all four components must be in the machine. The KIM-1 Monitor is in ROM and is always "in the machine". The PLEASE Executive resides in locations 1780 through 17E6 which is the "extra" RAM on the 6530 chips. This must be loaded from the cassette tape each time the system is powered up. It is the first program on the tape.

The PLEASE Functions reside in locations 200 through 3FF. These must be loaded from the tape, and are required for all of the programs.

The PLEASE Programs reside in locations 00 through 9F. Locations A0 through AF contain a table which decodes commands and is used to select which of several resident programs to run. Locations B0 through EE are used for a variety of PLEASE storage areas and are described elsewhere. The PLEASE Function table is in locations 100 through 11F. Special Functions have a table in locations 120 through 12F. Special Functions are located in locations 130 through 1EF. This section of memory must be filled from the cassette tape and changes from one collection of programs to another. In some instances, only the PLEASE Programs are changed and only locations 00-9F must be overlayed. Other programs require the overlaying of locations 00 through 1FF.

Load the Executive, PLEASE, and the Program Module you wish to run, following the instructions on page 4.

To start PLEASE, Set Address 178Ø and press "GO".

You are now in the Command Decoder which is waiting for a Command. A Command is issued by typing at least the first two characters of the Program Name and then pressing the "GO" key. For example, to run the STIMER (Set TIMER) program, type "S","T", and "GO". If STIMER has been properly loaded, it will be run. If it is not loaded, then you will remain in the Command Decoder until you issue a valid command. Once you start typing a command you can abort it, before typing "GO", by typing "PC" which will clear the Display and Command Buffer.

NOTE: The Command Decoder may be restarted at any time by pressing the "RS" (ReSet) key, setting the address to 178Ø if not the default, and then pressing the "GO" key.

Cassette Tape Loading Instructions

The reader is assumed to know how to load cassette tapes on the KIM-1. This section covers the specifics of loading PLEASE from the cassette tapes provided with the PLEASE package.

Tape Layout

The tape contains two copies of each of the PLEASE modules. Each set of modules consists of the individual modules with a ten second gap between modules. The tape was written using a variation of a program called SUPERTAPE developed by Jim Butterfield and presented in the KIM-1 User Notes, Volume 1, Issue 2, pages 12 to 14. SUPERTAPE produces tapes that are written at higher speeds than the standard KIM-1 tapes, but which are compatible and can be read using the standard KIM-1 monitor without any modification! The first set of PLEASE modules is written at six times the standard rate. The second set of PLEASE modules is written at the standard rate. These tapes were created by dumping via the modified SUPERTAPE directly from my KIM-1 to the cassette. The tapes were not copied. (I had tried having the tapes professionally copied, but that was an unqualified disaster). If you have a reasonably good cassette recorder you should be able to use the first, highest speed version. The other speed is included for anyone whose tape system is marginal. If even the standard, slow speed version, does not work with your equipment, then you have a problem. Best bet is to get/use another recorder and/or have the head alignment checked on your unit.

The sequence of modules in each set is:

Module Description	Tape ID Number
Executive	01
Timer/Clock/Notice/Daffy	02
Shooting Stars/Hi-Lo	03
Tipsy	04
Decimal/Hex Conversion	05
Adding Machine/Reaction Time	06

Bringing Up The System

When starting the system from scratch, the Executive should be the first module loaded. It loads into locations 1780 through 17E6. The second module loaded should be Timer/Clock/... which contains all of the basic PLEASE Functions, System Timer, Interpreter and so forth. At this point all of the programs in Module #1 can be run. The other program modules can then be loaded, as desired. These each overlay a portion of the basic PLEASE module.

Tape Input Instructions

```
Set Address   00F1  to  00   Set Binary Mode for Tape Input Routine
Set Address   17F9  to  00   To get NEXT module on the tape, or,
              17F9  to  ID   to get SPECIFIED module from the tape.
Set Address   1873           And press GO to start Tape Input Routine.
Start Cassette Tape Recorder.
Display will show  0000xx  at completion of a successful read.
```

Program Module #1

Contents:

ST	Set Timer	Set Initial Value for Millisecond Timer
DT	Display Timer	Display Millisecond Timer with Continuous Update of Timer
SC	Set Clock	Set Initial Value of 24 Hour Clock
DC	Display Clock	Display 24 Hour Clock with Continuous Update of Clock: Hours, Minutes, Seconds
NO	NOtice	Display an Alphabetic Message Six Characters at a Time
BB	BillBoard	Display an Alphabetic Message Shifting One Character at a Time from Right to Left.
DA	DAffy	A Sophisticated Number Guessing Game (also known as Mastermind)

Program Module #2

SS	Shooting Stars	The Shooting Stars Puzzle
HI	HI-lo	A Simple Number Guessing Game in which the Player tries to guess the number in the range 00 to 99 that the computer picked.

Program Module #3

TI	TIpsy	An Intoxication Tester in which the subject is shown a four digit decimal number for two seconds and after a one second delay tries to type in the number within ten seconds.

Program Module #4

DH	Decimal/Hex	Convert 1 to 6 digit decimal number to its hexidecimal equivalent. Range 00 to 999999.
HD	Hex/Decimal	Convert 1 to 6 digit hexidecimal number to its decimal equivalent. Range 00 to F423F

Program Module #5

AS	Add/Subtract	Six Digit decimal adding machine. Permits additions and subtractions as long as answer is in the range 00 to 999999. Does not permit entry of a number which will cause overflow or a negative result.
RE	REact	Reaction Time Tester. Time taken to react to a change in the Display is calculated and displayed with millisecond accuracy.

Set Timer
Display Timer

Program Module #1

Command: ST
Display: blank
Input: 0 to 6 digit decimal number. Will enter as rightmost digit and prior digits will be shifted one position to the left.
Type "PC" to erase value.
Type "+" to set value.
Action: Will automatically drop into the Display Timer Program.

Command: DT
Display: Current Value of the Millisecond Timer will be continuously displayed. The least significant digit displays the current millisecond value, the next digit the hundredths of a second, ..., and the most significant digit hundreds of seconds.

100	10	1	0.1	0.01	0.001	value in seconds
0	1	2	3	4	5	display position

Input: "GO"
Action: Return to PLEASE Command Decoder
or
Input: "+"
Action: Go to Set Timer Program

Notes and Comments:

1. The blank display in Set Timer has a value of all zero's. Therefore, calling ST and then typing "+" will start the Timer at zero. A "+" in the Display Timer restarts the Set Timer, a "+" in Set Timer restarts the Display Timer, so that two successive "+" inputs will restart the Timer at zero.

2. Although the Timer is accurate to a millisecond (give or take a few microseconds), the display appears to show a nine as the least significant (millisecond) digit. This is due to the Display Refresh algorithm, and is instructive in the workings of the dynamic display. Since the refresh moves one digit position each millisecond and there are an even number of digit positions, six, the least significant digit is changing every millisecond but is only displayed every sixth millisecond. This results in only the even or the odd values being displayed! In the Display Timer routine it happens to be the odd values: 1 then 7 then 3 then 9 then 5 then 1 ... if the original millisecond value from Set Timer is even. The segments which are used by the odd digits never use the lower left vertical display segment but do use all of the other segments at least twice. This causes the apparent "9" to be seen as the millisecond digit. If, however, the original millisecond value from Set Timer is odd, then the millisecond digit appears to be an eight. This is due to the fact that the segments used by the even digits are equally distributed and turn on all segments which appears as an eight!

Set Clock
Display Clock

Program Module #1

Command: SC
Display: blank
Input: Time of Day as Hour, Minute, and Second. This is a 24 hour clock so that 1:00 pm is represented as 13:00, 2:00 pm as 14:00,..., and 11:00 pm as 23:00. Input digits enter as left-most digit first, then second left most digit,..., rightmost digit last. Each field, Hours, Minutes, and Seconds is a two digit field. Any unfilled positions will be treated as zeros.
Type "PC" to erase value.
Type "+" to set clock value.
Action: Will automatically drop into the Display Clock Program.

Command: DC
Display: Current Value of the 24 Hour clock will be continuously displayed and updated every second. The least significant digit display the second, the next digit the tens of seconds, the next digit minutes,..., the most significant digit the tens of hours.

Hours 00 - 23 /	Minutes 00 - 59 /	Seconds 00 - 59	Value
0 1	2 3	4 5	Position

Input: "GO"
Action: Return to PLEASE Command Decoder
or
Input: "+"
Action: Go to Set Clock Program

Notes and Comments:

1. Compare the two methods of inputting decimal information in Set Clock and Set Timer. In Set Clock the digits are input from left to right with no shifting, similar to a typewriter, for example. In Set Timer the digits are input from right to left with shifting, similar to typical calculator.

2. The Clock and the Timer will continue to run as long as Program Module #1 is running. Incrementing of the Clock and Timer is performed as a part of the major Executive Loop and occurs whether or not the Set Timer and/or Set Clock routines have been called. Values of the Timer are often used by PLEASE to generate pseudo random numbers for various games such as DAFFY, HILO, and so forth.

3. The highest value for a 24 Hour clock is 23:59:59. One second later the clock reads 00:00:00.

Notice Billboard — Program Module #1

Command: NO
Display: A message will be shown on the display in Alphabetic, Numeric, and/or Special Characters. Each six character segment will be displayed for about one second. After the entire message has been displayed it will repeat. The basic message is:

```
PLEASE
 CAN I
 HELP
 ?
```

Input: "GO"
Action: Return to PLEASE Command Decoder
or
Input: "+"
Action: Go to Billboard Program

Command: BB
Display: A message will be shown on the display in Alphabetic, Numeric, and/or Special Characters. Characters will enter the display at the rightmost position and will then be moved across the display to the left a about a rate of 0.2 seconds per shift. This makes a moving billboard type of display. After the entire message has been displayed it will repeat. The basic message is:

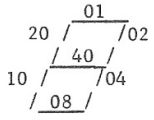
PLEASE CAN I HELP ?

Input: "GO"
Action: Return to PLEASE Command Decoder
or
Input: "+"
Action: Go to Notice Program

Notes and Comments:

1. You can write your own messages. The message starts at location 1DØ. Each memory location contains the code for the segments to turn on to represent a single character. Appendix A contains the hex values for the Alphabetic characters A to Z, the Numbers 0 to 9, and a few special characters. Or, invent your own characters. The message is terminated by a Hex FF code.

2. To create your own characters simply determine which segments are to be turned on. The Hex value for each segment is given in the diagram below.



Daffy Program Module #1

Command: DA
Display: blank
Input: Four digit decimal number starting at leftmost digit and filling "address" portion of the display. Range 0000 to 9999.
Type "PC" to erase guess.
Type "+" to have guess evaluated.
Action: If your guess does not exactly match the answer selected by the computer, then an evaluation will be displayed in the two rightmost display locations. The first digit will be the number of exactly correct matches you have made, that is, the digit is correct and is in the right location. The second digit indicates how many of your digits were correct without regard to position. For example:

Answer	Guess	Evaluation	
1976	1234	11	1 digit in correct location 1 digit correct without location
1976	1467	13	1 digit in correct location 3 digits correct without location
1976	4567	02	0 digits in correct location 2 digits correct without location

When your guess exactly matches the answer selected by the computer, then the number of guesses it took you will be displayed as a flashing number in the last two digit positions of the display.

1976	1976	15	flashing = it took you 15 guesses to get the answer.

Input: "GO" at any time
Action: Give up and return to PLEASE Command Decoder
or
Input: "+" after solving DAFFY
Action: Start a new round of DAFFY.
or
Input: "+" after an incorrect guess
Action: Clear old guess and get next guess.
or
Input: "PC" during input of guess
Action: Erase guess without having it evaluated or counted as a guess.

Notes and Comments:

1. Where does DAFFY get its answer? From the millisecond timer.

2. DAFFY is the PLEASE version of Mastermind which has been written up in several journals and other publications. The PLEASE version was written from scratch without "borrowing" from any of the other versions.

Shooting Stars Program Module #2

Command: SS

Display: The three leftmost display positions are used to display the Universe, which for Shooting Stars consists of a three-by-three array of "Stars". Initially, only the center star is in existance. The fourth position of the display is used to show your current shot, that is the number of the star you are shooting at. The last two display locations maintain a count of the number of shots you have taken. The initial display looks like:

```
. . .
. * .   where the * is a lit segment.
. . .
```

Input: Type number of star you wish to shoot. Stars are numbered as the bottom three-by-three section of the keyboard:

```
8 9 A
4 5 6
0 1 2
```

You may only shoot an existing star, that is one which is currently lit. Shooting a non-existant star will cause a question mark "?" to be displayed in the current shot display location, but will not be counted as a shot. A valid star shot will cause all stars in the domain of that star to vanish if lit, or to become lit if currently dark. The domain of each star is defined as follows:

```
. . .   . . .   . . .   8 . .   . 9 .   . . A   8 9 .   8 9 A   . 9 A
4 5 .   . . .   . 5 6   4 . .   4 5 6   . . 6   4 5 .   . . .   . 5 6
0 1 .   0 1 2   . 1 2   0 . .   . 1 .   . . 2   . . .   . . .   . . .
```

The object of the puzzle is to go from the initial state of only the center lit to the state where every star except the center is lit:

```
FROM:   . . .      TO:   * * *
        . * .            * . *
        . . .            * * *
```

Since in the initial state only star #5 is lit, your first valid shot must be at star #5.

Input: "+" (after selecting star number)
Action: Cause your current shot to be evaluated, the Universe modified, and the count to be incremented. If your shot is invalid, a question mark is displayed.
Input: "+" (after shot has been scored)
Action: Setup for next shot. Clear old shot number.
Input: "GO"
Action: Give up and return to PLEASE Command Decoder.
Input: "PC"
Action: Abort Current Shot without having it count.

Notes and Comments:

1. The optimal solution to the Shooting Stars puzzle requires eleven shots.

2. If you manage to wipe out all the stars in the Universe you lose, and will be informed by question marks in the count field of the display.

3. My first encounter with Shooting Stars was through an article by Willard I Nico, Delta t, 11020 Old Katy Rd., Suite 204, Houston Texas 77043, which appeared in the May 1976 issue of BYTE pages 42 through 49. PLEASE uses a modification of his "MASKTAB" which is used to test shots, map their effects and so forth. The rest of the code is unique to PLEASE. The table modification consisted of selecting a different set of position codes which made converting the Universe into the display segments for the KIM-1 display an easy task.

4. If you really get frustrated, can not figure out (or luck out) the solution, and if even your best friends won't tell you, then send a self-addressed, stamped envelop to me and I will supply the answer (which I got from a friend, to be perfectly honest!)

HiLo Number Guessing Game Program Module #2

Command:	HI
Display:	blank
Input:	Type two digit decimal number. Leading zero, if any, must be typed. Range of numbers is 00 through 99. The computer has selected a number within this range. It is your task, Jim, if you and the Mission Impossible...whoops, wrong show. It is your task to guess the computers number. On each of your guesses the computer will tell you if your guess was high (HI), low (LO) or correct (flashing number).
Input:	"+"
Action:	Get ready for next guess.
or	
Input:	"GO"
Action:	Return to PLEASE Command Decoder

Notes and Comments:

1. This game is included primarily for the kiddies. My own children, ages almost four and almost 6, really enjoy playing with the KIM-1, and through HILO are learning their numbers, both the single digits required for inputting their guesses, and the higher numbers which make up the answer.

2. HILO is written entirely in basic PLEASE functions. It can be easily modified to work with three or four digit numbers, or could be fixed to work with hexidecimal numbers.

Tipsy Program Module #3

Command:	TI
Display:	blank
Input:	"+"
Action:	Computer will select a pseudo random number and display it. The number is decimal, four digits, and in the range 0000 to 9999. The number will be displayed for two seconds. The display will then be blanked for one second. When two dash characters appear in the rightmost two digit positions of the display, the computer is ready to accept your answer.
Input:	Four digit decimal number and "+"
Action:	Digits will be displayed as they are typed in. At the plus input the computer will evaluate your answer. If it is correct, then the time it took you to answer, in tenths of a second, will be displayed. If your answer is incorrect or if you took more than ten seconds, then two question marks will be displayed.
Input:	"+"
Action:	Restart TIPSY with a new random number
or	
Input:	"GO"
Action:	Return to PLEASE Command Decoder.

Notes and Comments:

1. You may erase your answer and try again using the "PC" key. This will erase the digits you have typed, but the ten second clock is still running until you press the "+" key.

2. TIPSY is patterned after the "drunk" test which was part of an article "KIM-1 Microcomputer Module" by T. E. Travis in Microtrek, August 1976, pages 7 through 16. This article was very helpful in clarifying a number of important points about the operation of the KIM which are not well documented in the KIM-1 Manual, and it provided source listings for such useful operations as refreshing the display, getting a character from the keyboard, and so forth.

Decimal/Hex Conversion Program Module #4
Hex/Decimal Conversion

Command:	DH
Display:	blank
Input:	Type the decimal number that you wish converted. Leading zeros are not required. The numbers typed will appear in the right-most display position and all previous number will be shifted one position to the left. The range of valid decimal numbers is 000000 to 999999. The current input may be erased by typing a "PC", or you can just keep typing until you get the correct value entered as the numbers will keep shifting. Terminate the input by typing "+".
Action:	Decimal value in display will be replaced by its hexidecimal equivalent.
Input:	"+"
Action:	Restart Decimal/Hex conversion routine.
or	
Input:	"GO"
Action:	Start Hex/Decimal conversion routine.

Command:	HD
Display:	blank
Input:	Type the hexidecimal number that you wish converted. Leading zeros are not required. The range of valid hex numbers is 000000 to F423F. A value greater than F423F will cause all digit positions to be filed with question marks. Input is terminated by "+" or may be erased by "PC".
Input:	"GO"
Action:	Restart Hex/Decimal conversion routine.
or	
Input:	"+"
Action:	Start Decimal/Hex conversion routine.

KIM-1/6502 Information

Articles about the KIM-1 and the 6502 have appeared in many of the club newsletters. The following three references have shown particular interest in these MOS Technology products:

KIM-1/6502 User Notes, c/o Eric C. Rehnke, 425 Meadow Lane, Seven Hills, Ohio 44131 is, as the title implies, devoted to these products. Subscription is $5.00 for issues #1 thru #6. #1 was published in Sept 1976.

The COMPUTERIST, P.O. Box 3, S Chelmsford MA 01824, while being aimed at general microcomputer activity in the New England region, has shown a definite interest in the KIM-1, which is not suprising since the author of PLEASE is also the Editor/Publisher of The COMPUTERIST. Subscription is $6.00 per year or trade for something of equal value.

Personal Systems, official publication of the San Diego Computer Society, P.O. Box 9988, San Diego CA 92109, has shown KIM-1/6502 interest and in a recent issue cross referenced all the articles appearing in other publications dealing with 6502 based systems. Membership is $10.00 per year.

Add/Subtract Program Module #5

Command:	AS
Display:	blank
Input:	Type first decimal value, range 000000 to 999999. Leading zeros are not required. Terminate input with "+".
Action:	First number is stored away.
Display:	blank
Input:	Type second decimal value, range 000000 to 999999. Type "+" to add values. Type "DA" to subtract values.
Action:	Second value will be added or subtracted from the first value.
Display:	The result will be displayed. If the result would be greater than 999999 or negative the second value will be flashed to indicate an error condition.
Input:	"GO"
Action:	Return to PLEASE Command Decoder
or	
Input:	"+"
Action:	After valid operation: Accept new input to combine with the current total to provide a "chained" operation. After invalid operation: Clear total and accept new values.

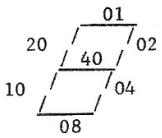
Notes and Comments:

1. Typing "PC" will erase the input and permit a new number to be entered.

Character Generator Table

Char.	Upper	Lower	Char.	Upper	Lower
A	77		U	3E	1C
B		7C	V		
C	39	58	W		
D		5E	X		
E	79		Y	6E	
F	71		Z	5B	
G	3D	6F			
H	76	74	Ø	3F	
I	30		1	06	
J	1E		2	5B	
K			3	4F	
L	38	06	4	67	
M			5	6D	
N	37	54	6	7C	
O	3F	5C	7	07	
P	73		8	7F	
Q		67	9	6F	
R		50			
S	6D		?	53	
T	07	78	-	40	

Notes: Some characters simply can not be made with the seven segment LED. One of the reasons for naming this demo package "PLEASE" was that the name looks real good on the display!



React Program Module #5

Command:	RE
Display:	blank
Input:	Type any character to start the test.
Action:	Test will be started.
Display:	All zeros will be displayed for a random period of time between 0.1 and 9.9 seconds in duration. After this time the display will be blanked.
Input:	Type any character as quickly as possible.
Action:	Time taken to type a character after the display was blanked will be calculated.
Display:	Time taken to react will be shown. The least significant digit displays milliseconds, the next digit hundredths of a second,..., and the most significant digit hundreds of seconds (in case you are taking a coffee break when the display goes blank).
Input:	"+"
Action:	Start new reaction time test.
or	
Input:	"GO"
Action:	Return to PLEASE Command Decoder.

Notes and Comments:

1. Your reaction time will depend on a number of factors. After seeing how fast you can be with total concentration, try different conditions such as:

   Counting backward from 100
   Reciting the alphabet backwards
   Singing
   Talking to someone

2. If you get 000003 for a time, it means that you pressed the key before the display went blank. Sorry.

The Last Word

I have worked very hard to make PLEASE a quality package. Since I did not have access to any sort of assembler, it has taken a tremendous amount of time to put this package together. I have tried to provide a good, clean set of documentation. My labors are justified if two events occur: first, that you and other users of PLEASE are entertained, educated, and perhaps challenged to do even greater things with your "humble" KIM-1 system, and, second, that you and other computer hobbyists permit me to obtain some financial return for my efforts by <u>not</u> permitting others to make copies of the PLEASE documents and cassette tapes. I have priced everything low so that no one would be forced to "rip-off" his copy due to an inability to pay. If, however, there is someone who can not afford to purchase PLEASE even at the low price, then please contact me. I will arrange some swap or exchange or something.

Command Summary Chart

Command	Purpose	Module	Page
ST	Set Millisecond Timer Initial Value	#1	6
DT	Display Current Value of Millisecond Timer	#1	6
SC	Set 24 Hour Clock Initial Value	#1	7
DC	Display Current Value of 24 Hour Clock	#1	7
NO	Display Notice at Six characters per frame	#1	8
BB	Display Notice in Moving Billboard fashion	#1	8
DA	Daffy: A Sophisticated Number Guessing Game	#1	9
SS	Shooting Stars Puzzle	#2	10
HI	HiLo: A Simple Number Guessing Game	#2	11
TI	Tipsy: An Intoxication Tester	#3	12
DH	Decimal to Hexidecimal Conversion	#4	13
HD	Hexidecimal to Decimal Conversion	#4	13
AS	Add/Subtract Calculator	#5	14
RE	Reaction Time Tester	#5	15

"GO"	In Command Decoder: Execute Command In Program (generally): Return to Command Decoder
"+"	In Program: Execute Next Program Step
"PC"	Always: Clear Current Input from Display
"DA"	In Add/Subtract: Subtract
"RS"	Always: Interrupt and Go To KIM Monitor

	GO	ST	RS		GO	ST	RS		
	AD	DA	PC	+	AD	DA	PC	+	
KIM	C	D	E	F	r	S	t	Y	PLEASE
Keyboard	8	9	A	B	L	n	o	P	Keyboard
	4	5	6	7	E	F	H	I	Alpha-Mode
	Ø	1	2	3	A	b	c	d	

Some Important Numbers

ØØF1	Status Byte: Must be set to ØØ when loading from tape.
17F9	Tape ID Number: Must be set to ID Number of tape segment being requested, or to ØØ for next segment on tape.
1873	Tape Input Start Address: Set Address and "GO"
ØØØØ	Tape Read Correctly Indicator.
FFFF	Tape Read Incorrectly Indicator.
17FA	"ST" Interrupt Vector: Must be set to ØØ to enable "ST".
17FB	"ST" Interrupt Vector: Must be set to 1C to enable "ST". When "ST" is enabled, Single Step Mode is also enabled.
178Ø	PLEASE Start Address: Initiate PLEASE and Go To Decoder.

617/256-3649 Where to call if a real problem occurs.