



MM5765 calculator programmer general description

The MM5765 provides a convenient and inexpensive means of adding "learn mode" programmability to many National Semiconductor calculator chips. It interfaces directly by simply adding a single static switch, four dynamic keys and a mean of displaying an alarm condition. The monolithic MOS integrated circuit combines P-channel enhancement and depletion mode technologies to obtain low voltage and low power characteristics necessary for economical battery-powered products.

The MM5765 is a dynamic key sequence programmer that memorizes any combination of key entries while in the Load Mode, then automatically plays back the programmed sequence as often as desired in the Run Mode. Up to 102 characters can be stored in multiprogram sequence blocks. Each block, or program, can be executed individually or the operator can make the decision to branch to specific programs, run each in series or perform intermediate calculations from the keyboard. When programming in the Load Mode, the Delete key provides a convenient editing feature and the Halt key programs variable data entry points where control is temporarily returned to the operator in the Run Mode. Start and Skip keys control operation in both modes.

Synchronization with the calculator chip is accomplished by monitoring its Digit Output and Ready signals. The digit signals give timing information while the Ready indicates status of the calculator and synchronizes the key entry interface between it and the MM5765.

Up to four switch inputs (K1, K2, K3 and K4) and up to twelve digit lines are connected in parallel with the calculator switch and digit terminals that scan the keyboard. Keys stored in the MM5765 that are entered by selecting K1 through K4 are encoded simply as matrix positions, i.e., a particular switch input at a specific digit time. Therefore it is the key matrix address that is stored and not the key function. (Con't on page 4)

features

- Many NSC calculator chips can be provided with programming capability with the addition of only one static switch and four dynamic keys.
- Any key sequence, including constants and date entry points, may be stored automatically in the Load Mode and executed in the Run Mode.
- 102 step storage capacity of up to 47 different keys arranged in a 12 x 4 matrix.
- Multiprogram capability
- Provision for editing in Load Mode using the Delete key
- Convenient verification of programs using a Step Mode feature
- Alarm for full storage condition—or if a deletion of the first step in a program is attempted
- Power-on clear
- Direct 9V battery compatibility

block and connection diagrams

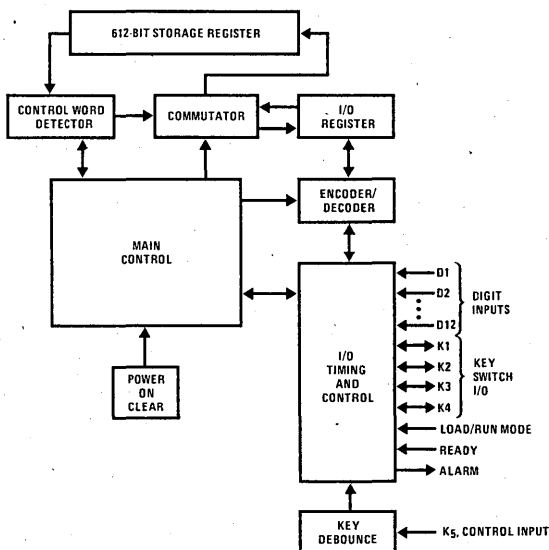
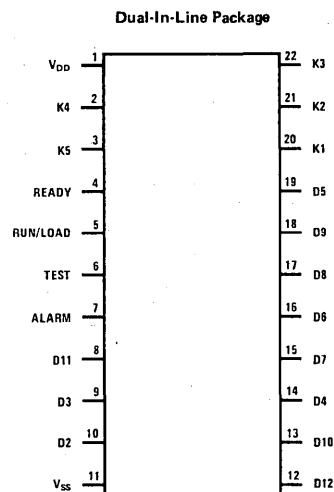


FIGURE 1



TOP VIEW
Order Number MM5765N
See Package 21

absolute maximum ratings

Voltage at Any Pin Relative to V_{SS} $V_{SS} + 0.3V$ to $V_{SS} - 12V$
 (All other pins connected to V_{SS})
 Ambient Operating Temperature $0^{\circ}C$ to $+70^{\circ}C$
 Ambient Storage Temperature $-55^{\circ}C$ to $+150^{\circ}C$
 Lead Temperature (Soldering, 10 seconds) $300^{\circ}C$

operating voltage range
 $V_{SS} - 6.5V \leq V_{DD} \leq V_{SS} - 9.5V$

(V_{SS} is always the most positive supply)

dc electrical characteristics

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Operating Supply Current (I_{DD})	$V_{DD} = V_{SS} - 9.5V$ $T_A = 25^{\circ}C$		8.0		mA
Keyboard Scan Input Levels (K1, K2, K3, K4, K5)					
Logical High Level (V_{IH})	$V_{DD} = V_{SS} - 6.5V$	$V_{SS} - 2.5$		$V_{SS} - 5.0$	V
Logical Low Level (V_{IL})	$V_{DD} = V_{SS} - 9.5V$			$V_{SS} - 6.0$	V
Digit Input Levels (D2 through D12)					
Logical High Level (V_{IH})	$V_{DD} = V_{SS} - 6.5V$	$V_{SS} - 2.5$		$V_{SS} - 5.0$	V
Logical Low Level (V_{IL})	$V_{DD} = V_{SS} - 9.5V$			$V_{SS} - 6.0$	V
Other Inputs (Ready, Run and Test)					
Logical High Level (V_{IH})		$V_{SS} - 2.5$			V
Logical Low Level (V_{IL})	$V_{DD} = V_{SS} - 6.5V$ $V_{DD} = V_{SS} - 9.5V$			$V_{SS} - 5.0$ $V_{SS} - 6.0$	V V
Switch Buffer Output Levels (K1, K2, K3, K4)					
Logical High Level (V_{OH})	$V_{DD} = V_{SS} - 6.5V$	$V_{SS} - 1.5$		V_{SS}	V
Logical Low Level (V_{OL})	$V_{DD} = V_{SS} - 9.5V$			$V_{SS} - 6.0$ $V_{SS} - 7.0$	V V
Alarm Output Current Source Current	$V_{OUT} = V_{SS} - 4.5V$, $V_{DD} = V_{SS} - 6.5V$ $V_{OUT} = V_{SS} - 5.2V$, $V_{DD} = V_{SS} - 7.25V$ $V_{OUT} = V_{SS} - 7.8V$, $V_{DD} = V_{SS} - 9.5V$	-5.0	-8.0	-20.0	mA mA mA

ac electrical characteristics

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Digit Input Time (Figure 3)		70			μs
Word Time (Figure 3)		0.64			ms
Switch Input Time (Figure 3)		0.70			μs
Switch Output Time (Figure 4)		70			μs
Switch Propagation Delay Output (Figure 4)			15	26	μs
Switch Output Transition Time (Figure 4)	$C_{LOAD} = 100 pF$		2		μs
Switch Input K5 Key Bounce-out Stability Time (The time a keyboard input must be continuously higher than the mini- mum Logical High Level to be ac- cepted as a key closure, or lower than the maximum Logical Low Level to be accepted as a key release, i.e., 6 or 7 cycles of D2.)		4.5		17.0	ms
Key Closure Rate (Time between consecutive key outputs in Run Mode.)			40		ms
Key Acceptance Rate (Time between consecutive key inputs in Load Mode.)				47	ms

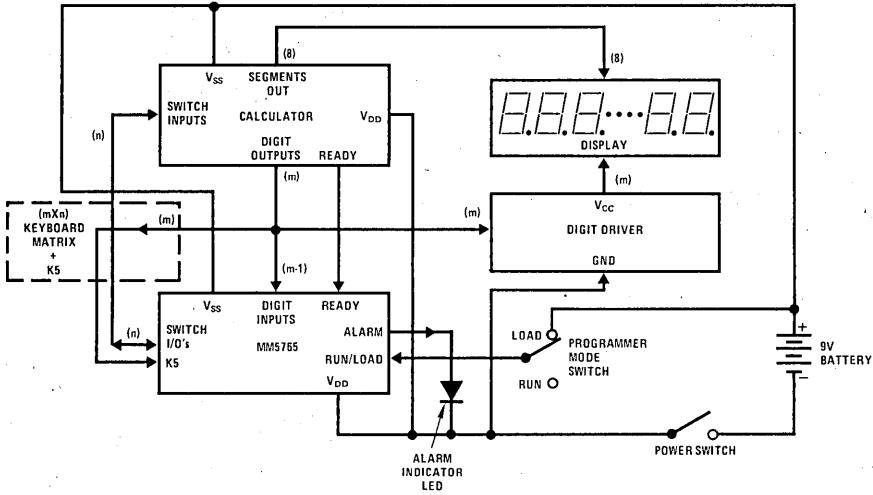


FIGURE 2. MM5765 Programmer Connected in Low-Cost Battery Operated Calculator System

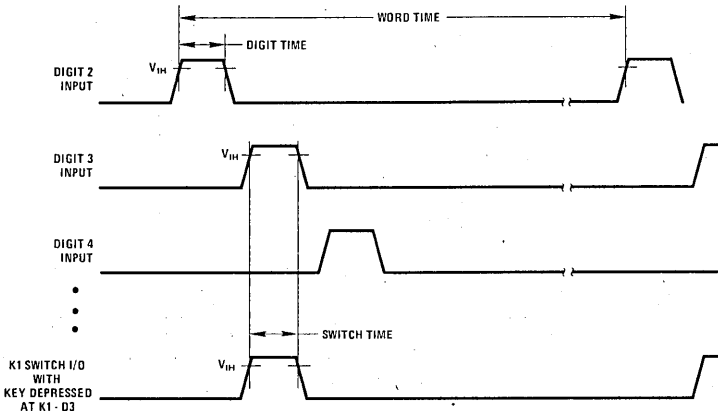


FIGURE 3. Input Timing

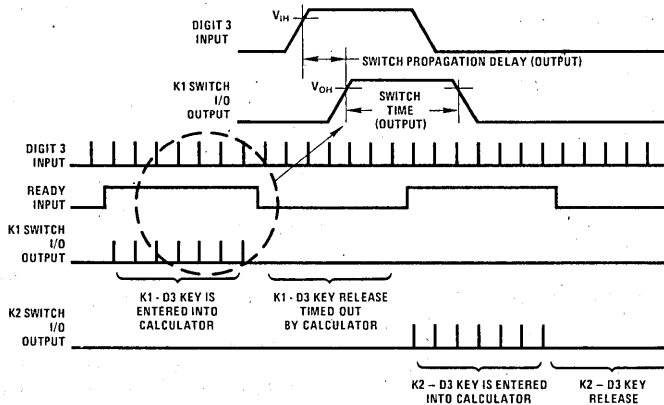


FIGURE 4. Programmer Output Timing

TABLE I. Action of Dynamic Control Keys as a Function of The LOAD/RUN Mode

KEY	LOAD	RUN
START	Clears and initializes program storage area.	Starts program when stopped in <i>HALT</i> mode. Starts first program.
SKIP	Terminates current program and initializes a new one.	Skip remainder of current program and begins execution of next one.
HALT	Programs an operator data entry or check point in <i>RUN MODE</i> .	
DELETE	Erases the last key entered. (Acts as a backspace key.)	

general description (con't)

Forty-seven different addresses can be stored using a 12x4 keyboard matrix. (The illegal address is Digit 1 and K4.) Switch Input K5 is used to enter programming control signals only and is not connected to the calculator in any way. The K5 input has key debounce protection identical to the calculator chip, which debounces K1 through K4. The MM5765 does not accept a K1, K2, K3 or K4 input until the Ready signal from the calculator goes from an idle, or high state, to a low state—indicating the key has been debounced by the calculator.

The program chip is dynamic, which means power must not be interrupted if a program is to remain stored. When power is applied an internal circuit automatically clears the MM5765, inhibiting false entries to the calculator and conditioning the system for entry of a new program.

Actual storage of the sequential key information is in a 612-bit shift register (see *Figure 1*). Each input character is encoded into a six-bit word and placed in the I/O register. If a Ready input confirms the character has been accepted by the calculator as a valid key entry, or the internal key debounce circuit in the case of Switch Input K5, the new key information is transferred by the commutator to the storage register. It is always placed in sequence at the end of the existing program, and an internal pointer is advanced six bits. The control word detector keeps track of the pointer and special codes required for control and alarm situations. In the Run Mode, characters are sequentially transferred into the I/O register, decoded on command of the Ready signal and entered into the calculator via the appropriate Switch Input Line.

When the MM5765 is used with calculators with long execution times, it may be useful to use a buffered Ready signal to drive a "Busy" indicator. This would give the user a visual feedback of status during Run operations.

PROGRAMMER CONTROL FUNCTIONS

"Load/Run" Mode Control

This control requires a single-pole, single throw static switch. It prepares the MM5765 for either accepting a key sequence or playing it back. Its position controls the function of the dynamic keys as shown in Table I.

Additional steps or programs can be appended to a stored key sequence even after execution simply by

switching back to the Load Mode and entering the new steps. The storage register pointer always returns to the end of the previously entered key sequence when the mode is changed from Run to Load, and to the beginning of the first program when changed from Load to Run.

"Start" Key (Refer to Table II for keyboard connections)

The function of this dynamic key depends upon the position of the Load/Run Mode Control Switch:

1. With the Mode Switch in the Load position, Start clears the entire program storage register of all programs and initializes the device for accepting a new set of programs by setting the pointer at the first storage location.
2. With the Mode Switch in the Run position, Start begins execution of the first program, or if pausing in the Halt Mode, continues the program. This key is not seen by the calculator and therefore has no affect on the calculations in progress.

The Start key is timed out by the key bounce-out stability timer of the MM5765 on both key entry and release.

TABLE II. Control Signal Input, K5, Keyboard Matrix

CONTROL KEY FUNCTION	DIGIT TO K5 CONNECTION
START	D5 to K5
SKIP	D6 to K5
DELETE	D7 to K5
HALT	D8 to K5

"Skip" Key

This is the other dynamic key whose function depends on the position of the Load/Run Switch:

1. In the Load Mode, this key terminates the current program and marks the beginning of a new program. Repetitious depressions will be ignored. The Delete key will erase this key from the storage register, but the Alarm will be set indicating to the user that a

complete program has been deleted. A new Skip will reinitiate the deleted program; otherwise, subsequent deletions or additions will be to the previous program.

- In the Run Mode, if the MM5765 is at a Halt, the Skip key will cause the remaining steps of the current program to be skipped. Execution automatically begins again at the start of the next program and continues to the first programmed Halt; in the absence of a Halt, execution will continue to the end of the program.

Depression of this key is not seen by the calculator and does not affect its status. The Skip key is timed out by the key bounce-out stability timer of the MM5765 on both key entry and key release.

"Halt" Key

The Halt key is a dynamic key that has a function only in the Load Mode. It is ignored in the Run Mode.

The Halt key is used to program a data entry pause in the playback of a key sequence. When a Halt occurs in the program sequence during operation in the Run Mode, the MM5765 ignores all key entries except Start or Skip. The calculator chip accepts all nonprogrammer keys in the normal manner so that constants or variables can be entered, or intermediate calculations can be performed. The operator may use the Halt as a decision making point where he has the option to continue the program in a number of ways based on an intermediate result; e.g., skip to another program, restart the present program, or even go to a co-routine in a second MM5765 program chip.

If the user switches to the Load Mode during a Halt, execution of the current program will be terminated and the MM5765 will be ready to store additional keys at the end of the last program. If the mode is then returned to Run, Start will begin execution at the beginning of the first program.

The Halt key is debounced by the MM5765.

"Delete" Key

The Delete is another dynamic control key that functions only in the Load Mode and is ignored in the Run Mode.

It provides a method of editing by erasing the end step of the program. It is essentially a "backspace" key. Multiple Deletes can be used to remove several steps or even complete programs, but the Alarm will be set

if a Skip code is deleted or an attempt is made to delete the Start code (beginning of first program).

The Delete key is debounced by the MM5765.

Switch Input K5 Keyboard Bounce Protection

The MM5765 programmer chip is designed to interface with most low-cost keyboards and has characteristics identical to the standard NSC calculator keyboard bounce protection circuits.

A control key closure is sensed when Switch Input K5 is forced more positive than the Logical High Level specified in the Electrical Specifications. At the instant of closure, an internal "Key Bounce-out and Stability Time" counter is started. Any significant voltage perturbation occurring on the K5 input during timeout will reset the timer. Hence, a key is not accepted as valid until noise or ringing has died out and the stability time counter has timed-out. Noise that persists will inhibit key entry indefinitely. Release is timed in the same manner. The actual control operation is performed by the MM5765 after the release is validated, to differentiate the action from a calculator key.

ALARM CONDITIONS

An alarm condition will be indicated by the MM5765 program chip as a Logical High Level output on pin 7. An alarm condition can exist due to three circumstances:

- All 102 storage locations in the storage register are full. The Alarm is reset by entering a Delete key or if the mode is changed to Run and any key is pressed. When the storage register is full, subsequent data keys are ignored; the existing program is not disturbed.
- An attempt is made to delete a Start key code in the storage register during editing of a program. The alarm is set and the Delete key is ignored. Any of the calculator keys, the Skip or Halt keys or moving the Mode Switch to Run and pressing any key will reset the Alarm.
- A Skip key code is deleted from the storage register while editing. The alarm is set and the Skip is deleted. Any calculator or programmer key, or switching to the Run Mode and pressing a key will reset the alarm condition. If a Skip key is not re-entered, new key entries will be appended to the previous program, and the original program being edited will no longer exist.

TABLE III. Ready Signal Description

CALCULATOR FUNCTION	READY SIGNAL
Idle	Ready is quiescently at a Logical High Level ($\sim V_{SS}$).
Key entry and functional operation	When a key is depressed, the calculator bounce-out stability timer is initiated. Ready remains high until the bounce-out time is completed and the key is entered, at which time it changes to a Logical Low Level ($\sim V_{DD}$).
Key release and return to idle	Ready remains low until key release is debounced and the calculator returns to the idle state. The low to high transition signals the return to idle.

TABLE IV. Mode and Alarm Truth Table

PIN	MODE	LEVEL
Load/Run Input	RUN	LOW
	LOAD	HIGH
Alarm Output	ACTIVE	HIGH
	INACTIVE	LOW

TYPICAL OPERATION

Loading a New Program

At power-on, the MM5765 automatically clears and initializes the storage register. All that is necessary to start programming is to switch to the Load Mode. If unwanted programs already exist in the storage register from previous operations, switching to the Load Mode and depressing Start will clear the memory and initialize a new program.

Programming is accomplished by simply keying the calculator in the normal manner. The MM5765 memorizes each key in the sequence entered. It is usually convenient to have the calculator displaying as the program is entered to catch entry errors and keep track of progress. However, it is necessary to consciously consider the anticipated results when programming to ensure a meaningful display at each step. For example, wherever variables are to be entered in the program, the Halt key is used rather than any numeric value. Because the calculator chip does not see a Halt, the display will no longer be correct as the remainder of the sequence is loaded. One convenient way around the problem is to depress and hold the Halt key down while a dummy variable is entered into the calculator. The depressed Halt key will lock-out the MM5765 without affecting the calculator. An alternate approach would be to enter the Halt and the dummy variable, followed by the proper number of Delete keys required to erase the dummy variable from the storage register. Either approach results in a valid calculator display and stored program during programming.

Because the primary reason for using a key sequence programmer is to allow convenient recall of often used routines or in optimizing a particular solution by iterating a function many times with a variety of input variables—in other words, many iterations of a common sequence—it is always worth the time to spend a few minutes planning the best way of entering the program. Learning what the calculator should be displaying at each step of the programming can be done conveniently by keying the program while in the Run Mode, using the proper dummy variables, and jotting down intermediate results. In this manner potential calculator overflow conditions are caught, and subsequent Load Mode entry errors can be easily detected. When an entry error is made while programming in the Load Mode, use the Delete key to erase as many steps as necessary, switch back to the Run Mode and depress Start to correct the calculator display and return to the Load

Mode to finish. If the program does not approach the 102 key capacity of the MM5765, you may wish to simply use the calculator functions (such as Clear Entry) to correct the error situation even though they will be included in the stored program.

When the program is correctly loaded move the Mode Switch to Run. The program is now ready to be executed. Additions can be made to the program (even after execution in the Run Mode) by returning to Load. New key entries will be automatically appended to the end of the existing stored sequence. By executing the program before returning to Load, the calculator display will have a valid display and be in the correct state for properly displaying the new key additions. In this manner long programs may be constructed by connecting together a series of short sequences which are debugged as you go (reducing the possibility of error and minimizing confusion).

Running a Program

Use of a stored program requires only that the calculator be preconditioned, if necessary, and the Start key depressed while in the Run Mode. The program will continue to the end, or until a Halt is encountered in the key sequence.

Halts act as a pause during execution to permit entry of variable data, manual calculation of data, or checking of intermediate values. They are also available as user decision points for jumping to subsequent programs and can provide the capability for multiprogram labeling. When a Halt is encountered during execution, the MM5765 stops making key closures and returns control to the keyboard.

Upon reaching the end of a program, the internal pointer will return to the beginning and wait for another Start key.

As discussed above, programming certain sequences can result in errors in the calculator chip either during loading or during execution. If an error occurs as the program is loaded, the MM5765 will continue to store key depressions as they are made—independent of the calculator. Such a situation exists if a calculation results in overflow during execution of a stored program. The MM5765 continues to step through the sequence completely independent of calculator status as long as the Ready signal responds properly.

Multiple Programs

Use of the Skip key in the Load Mode codes that location as the beginning of a new program, just as the Start key is used to initialize the first program. All other aspects of loading the program are the same.

When a program stops at a Halt during execution, the user has the option of pressing the Skip key to jump to the next program or the Start key if he wishes to continue the original sequence. When control passes to the next program, execution begins and proceeds to the end of that program or until a Halt is encountered.

This property of automatically executing a program down to the first Halt provides a convenient method of labeling multiprograms. For example, entering a program with the sequence:

Start	(Load/Run = Load Mode)
1	
Halt	
C	(Calculator Clear Entry)
⋮	
⋮	Desired key sequence for Program No. 1
Halt	
Skip	
2	
Halt	
C	
⋮	
⋮	Desired key sequence for Program No. 2
Halt	

has stored two program sequences. In the Run Mode, pressing Start will display a "1", a second Start will execute Program 1 (or to the first internal Halt) eventually stopping at the last Halt and displaying a program result. The operator now has the opportunity to make a decision. He may rerun Program 1 by using the Start key, or continue to Program 2 by depressing the Skip key.

If he chooses Skip, a "2" will be displayed indicating that Program 2 has been addressed (as programmed by the Skip-2-Halt sequence at the beginning of Program 2 in the Load Mode). Start will then execute Program 2 down to its first Halt. The Program 2 result can be displayed by inserting another Halt at the end of that sequence. If a third program has been stored in the MM5765, depressing Skip will move the internal pointer to the beginning of that program and execute it to the first Halt. Assuming a Skip-3-Halt sequence was used at the front of the program, a "3" would be displayed by the calculator. If the operator had wished to rerun Program 1, instead of advancing to Program 3, he would have used Start (internal pointer is initialized), Start (displays shows "1") and Start (program is executed). For a rerun of Program 2 from the last Halt of Program 2, he would push Start (internal pointer is initialized) and Skip (pointer locates the top of Program 2, executes to first Halt and calculator displays "2").

Adding a Step Mode Feature

By returning the Ready input of the MM5765 to V_{SS} when the Mode Switch is in the Run Mode position, and depressing any of the control keys (Start, Skip, Halt or Delete) the program stored in the MM5765 may be executed and advanced one step at a time. This provides a convenient method of debugging programs.

Figure 5 shows the wiring of a 2-pole, 3-position switch used as the Mode Switch of a Programmer/Calculator system with the Step Mode as an added feature. Switching from the Load Mode to the Step Mode conditions

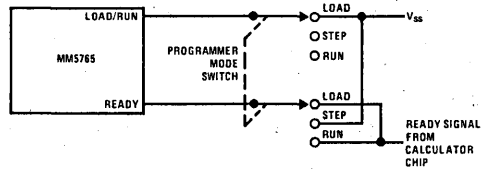


FIGURE 5. Switch Wiring for Adding Step Mode

the programmer to step through the stored program starting from the first entry of the first program. Start must be used to initiate the sequence, then any of the control keys can be used. Each depression of Start, Skip, Halt or Delete will advance the program being executed by the calculator one step. When a Halt is encountered in the program while in the Step Mode, the MM5765 ignores all key entries except Start or Skip just as described in Table I. If the Mode Switch is moved to Step, from a Halt point in the Run Mode, the program may be stepped from that point on by using Start or Skip followed by depressions of any of the control keys. Switching to Run from any intermediate point of a Run operation from that point. From a Halt, a Start or Skip Key must be pressed after switching to the Run Mode.

PROGRAMMING EXAMPLES

These examples assume use of the MM5738 calculator, which is an 8-digit, floating point, algebraic notation, single memory chip with constant operation. Please review the MM5738 data sheet for explanation of keyboard notation and function capability.

Example 1

A problem often encountered in communications design is the solution of

$$X = Y \left[\frac{\sin \theta}{\theta} \right]$$

With a programmer and even a simple calculator like the MM5738, this problem can be repetitively solved easily without tables. First, program the sequence for approximating $\sin \theta$ using

$$\sin \theta \cong \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!}, \text{ where } \theta \text{ is expressed in radians.}$$

$$= \frac{120\theta - 20\theta^3 + \theta^5}{120}, \text{ where } 5! = 120$$

$$= \frac{[(\theta^2 - 20)\theta^2 + 120]\theta}{120}$$

Example 1 (Con't)

KEY	DISPLAY	RUN/LOAD	COMMENTS
C		Load	
C	0	Load	Clear calculator
2	2	Load	Dummy variable "2" for θ is entered.
Start	2	Load	MM5765 is initialized.
MS	2	Load	
X	2	Load	
=	4	Load	θ^2 is formed
-	4	Load	
20	20	Load	
X	-16	Load	
MR	2	Load	
X	-32	Load	
MR	2	Load	
+	-64	Load	$(\theta^2 - 20)\theta^2$ is formed.
122	122	Load	122 is an entry error
C	-64	Load	After entering "C", operator can simply
Delete	-64	Load	continue by entering 120, or can correct
Delete	-64	Load	program sequence by deleting last four
Delete	-64	Load	keys. Result is the same, except the second
Delete	-64	Load	alternative would use less program storage.
120	120	Load	
X	56	Load	
MR	2	Load	
÷	112	Load	
120	120	Load	
=	0.9333333	Load	Sin θ for $\theta = 2$ radians is displayed

Check program by executing with $\theta = \frac{\pi}{4}, \frac{\pi}{3}$

3.14	3.14	Run	Enter approximation of π
÷	3.14	Run	
4	4	Run	
=	0.785	Run	$\theta = \frac{\pi}{4}$, in radians
Start	0.7068613	Run	$\sim \sin \frac{\pi}{4}$ displayed
3.14	3.14	Run	
÷	3.14	Run	
3	3	Run	
=	1.0466666	Run	$\theta = \frac{\pi}{3}$, in radians
Start	0.8660287	Run	$\sim \sin \frac{\pi}{3}$ displayed

Now we would like to add to the same program the rest of the expression:

$$Y \frac{\sin \theta}{\theta}$$

KEY	DISPLAY	RUN/LOAD	COMMENTS
1	1	Run	"1" is dummy variable for sin θ
Halt	1	Load	"Halt" is tagged onto end of existing program to allow readout of sin θ during execution
÷	1	Load	
MR	1.0466666	Load	
X	0.955414	Load	
Halt	0.955414	Load	Allows for Y entry
1	1	Load	Dummy variable for Y
Delete	1	Load	Dummy variable is removed from program by Delete, or Halt could have been held down while 1 is entered, in which case Delete would not be required.
=	0.955414	Load	

Example 1 (Con't)

Problems can now be solved using the program.

$$\text{Evaluate: } 0.54 \frac{\sin(0.72)}{0.72}$$

KEY	DISPLAY	RUN/LOAD	COMMENTS
0.72	0.72	Run	Enter $\theta = 0.72$ radians
Start	0.6594044	Run	Sin (0.72) displayed
Start	0.9158394	Run	
.54	0.54	Run	Enter variable Y
		Run	
Start	0.4945532	Run	$0.54 \frac{\sin(0.72)}{0.72}$ displayed

A sequence could easily have been included to convert degrees to radians.

PROGRAMMING

As an example of a multiprogram application, consider an automobile salesman who needs to calculate price plus sales tax, down payment and monthly payment on new cars many times a day. Again assume use of the MM5738 (although more powerful NSC calculators could obviously make the problem even easier). To simplify the example, assume the finance time is fixed at 36 months and the interest rate at 12% of the unpaid balance.

KEY	DISPLAY	RUN/LOAD	COMMENTS
C		Load	
Start		Load	Clear calculator and programmer. Label Program No. 1
1	1	Load	
Halt	1	Load	
C	0	Load	Clear program label.
5	5	Load	Sales tax = 5%.
%	0.05	Load	
X	0.05	Load	
Halt	0.05	Load	
100	100	Run	Load dummy variable for car price. Switching to Run is another method of entering a dummy variable without having to Delete.
MS	100	Load	
+	5.	Load	
K=	105.	Load	
Halt	105	Load	Program No. 1 displays price + tax amount.
Skip	105	Load	Initialize Program No. 1
2	2	Load	
Halt	2	Load	Label Program No. 2
C	105	Load	Clear program label.
Halt	105	Load	"Halt" for down payment %.
20	20	Run	Dummy down payment %.
%	0.20	Load	
X	0.20	Load	
MR	100	Load	
=	20	Load	
Halt	20	Load	Program No. 2 displays required down payment.
Skip	20	Load	Initialize Program No. 3.
3	3	Load	
Halt	3	Load	Label Program No. 3.
C	20	Load	Clear program label.
-	20	Load	
MR	100	Load	
=	-80	Load	Program No. 3 computes monthly payment from equation
MS	-80	Load	
1.01	1.01	Load	
X	1.01	Load	Monthly payment = $[\text{Total loan } (1 + i/q)^{nq}/nq]$
1.01	1.01	Load	$i = \text{interest per year, 12\% is assumed.}$
=	1.0201	Load	$nq = \text{total number of months} = 36$
			$q = 12 \text{ months per year}$
			$(1 + i/q) = 1.01$

PROGRAMMING (CON'T)

KEY	DISPLAY	RUN/LOAD	COMMENTS
=	1.040604	Load	(1.01) ⁴
=	1.0828566	Load	(1.01) ⁸
=	1.1725784	Load	(1.01) ¹⁶
X	1.1725784	Load	
1.01	1.01	Load	
=	1.1843041	Load	(1.01) ¹⁷
K=	1.1961471	Load	(1.01) ¹⁸
=	1.4307678	Load	(1.01) ³⁶
X	1.4307678	Load	
MR	80	Load	
÷	114.46142	Load	
36	36	Load	
=	3.179483	Load	
Halt	3.179483	Load	Program No. 3 displays required monthly payment.

EXECUTION OF PROGRAM

Salesman has potential customer for \$4995.95 automobile. Bank requires 20% down. The customer wants to know amount of down payment and monthly payments over 3 years at 12%.

KEY	DISPLAY	RUN/LOAD	COMMENTS
Start	1	Run	Program No. 1 label.
Start	0.05	Run	Sales tax displayed.
4995.95	4995.95	Run	Price entered.
Start	5245.7475	Run	Price + tax displayed.
Skip	2	Run	Program No. 2 label.
Start	5245.7475	Run	
20	20	Run	Enter % down.
Start	999.19	Run	Down payment displayed.
Skip	3	Run	Program No. 3 label.
Start	1588.4543	Run	Monthly payment displayed.