

LINE INPUT with timeout

This module is part of the original MMBasic library. It is reproduced here with kind permission of Hugh Buckle and Geoff Graham. Be aware it may reference functionality which has changed or is deprecated in the latest versions of MMBasic.

Getin.bas

```
'TIMER CONTROLLED LINE INPUT ROUTINE
'By Grogster of TBS forums 2013, with cursor
'contributions by TassyJim of TBS forums.

'-----

'This code can be called as a subroutine whenever you need
'the equivalent of LINE INPUT, but with a timeout feature.
'This routine will also ignore and exit with anything OTHER
'then numbers and letters, so special characters etc, will cause
'the subroutine to end. A flag is set during this process, so that
'exit from illegal keypresses can be detected by the main program.

'The code was originally developed for use in a menu-driven program
'where it was necessary that if no key was pressed when in any
'LINE INPUT type command, the routine could auto-exit back to the
'main menu if the set time for entering the data had expired.

'The routine also has the ability to show the output string as it
'is being typed, or to suppress outputting the keystrokes, which
'is very handy if you are wanting the user to enter a password.

'Backspace is fully processed, and any error in typing where that
'is corrected using the backspace key, is allowed for, so that
'only the wanted output is the result.

'Several variables are passed to the routine when it is called, and
'resulting flags etc are also set or reset during the routine.
'There is no allowance for incorrect data in the variables passed
'to the routine when it is called.

'To call the routine, use the line 'GETIN (CX,CY,TP,SK)' in
'your code(without quotes, naturally), where you want the
'input from the user.

'CX - X screen position of cursor prompt
'CY - Y screen position of cursor prompt
'TP - Time Period in ms before the timeout occurs
'SK - Show Keystrokes or not(1=show keystrokes, 0=don't show keystrokes)

'Any complying keypress will reset the timeout timer, so that if you have
```

'TP set to 5000(5 seconds), then so long as you type one character after
'another within 5 seconds, the timer will be reset for the next keypress.
'If you don't press another key after 5 seconds(using example above), then
'the routine will auto-exit.

'To detect if the routine has sucessfully exited with normal and correct
'input or not, you need to check GIFLAG after the routine has exited.
'If GIFLAG=0, then the routine has returned with a valid input.
'If GIFLAG=1, then the routine has exited with an invalid keypress.
'This can be used to detect invalid inputs, and can loop to the relevant
'error trapping routines, if that is required.

'When the routine exits, if the input is invalid, then LIN\$ will equal ""
'If the routine exits with valid input, then that input will be held
'in the LIN\$ string variable, and this can be subjected to further
'testing, depending on your application or requirements for the data
'typed in by the user.

'Probably not the most beautiful or efficient of code in the world,
'but it works! ;)

'Grogster - 4th February 2014, 8:16PM
'BUG FIX FOR ENTER ONLY - 26/02/2014

'-----

SUB GETIN (CX,CY,TP,SK)

do

timer=0

flag=0

giflag=0

if SK=1 then

print @(CX,CY) out\$ + "_ "

else

print @(CX,CY) "_ "

endif

do

k\$=inkey\$

loop until k\$<>" or timer>TP

if timer>=TP then

out\$=""

giflag=1

exit sub

endif

if asc(k\$)>=0 and asc(k\$)<=7 or asc(k\$)>=9 and asc(k\$)<=12 then

out\$=""

giflag=1

exit sub

```
endif
if asc(k$)>=14 and asc(k$)<=31 or asc(k$)>=128 and asc(k$)<=156 then
  out$=""
  giflag=1
  exit sub
endif
if asc(k$)=13 then
  If out$="" Then
    lin$=""
    exit sub
  else
    lin$=out$
    flag=1
  endif
endif
if asc(k$)=8 and len(out$)>=1 then
  out$=left$(out$,len(out$)-1)
else
  out$=out$ + k$
endif
if len(out$)=1 then
  if asc(out$)=8 then out$=""
endif
if flag=1 then
  out$=""
  giflag=0
  exit
endif
loop
end sub
```

From:

<https://fruitoftheshed.com/wiki/> - FotS

Permanent link:

https://fruitoftheshed.com/wiki/doku.php?id=mmbasic_original:line_input_with_timeout

Last update: **2024/01/19 09:39**

