

Magic Switchboard

[magicbox.gif](#)

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.

Magic Switchboard

This is rewrite of the magic switchboard program to suit MMBasic. The original was written by Picaxe technical for the picaxe 18x.

This program uses four input switches connected to pins 11, 12, 13 and 14 and the output (either white LEDs with colour tops or 6V coloured lamps) connected to Pins 15,16,17 and 18. The lamps require driver transistors. The schematic of the Magic Switchboard is in the attached file and you can see a demonstration of it at www.youtube.com/watch?v=0IGP8nQLANU. It is truly amazing.

Regards Bill Brown

```
' Magic Switchboard - original source by "Technical" from PicAxe Forum
'                               Modified for PicAxe 18x by Wayne Thomason of Addison,
TX USA
'                               7/15/2009
'                               Modified for Maximite MMBasic by Bill Brown  bill.b
'                               19/01/2012
'
'                               mods: 1. Now is easily configurable via switch? and
bulb? variables
'                               2. "timeout" functions even without learning
all 4 switches
'                               3. starting point and sequence direction
dependent on last switch turned off
'                               4. Now has Audience_lockdown feature.  If power
is turned on while switch-4
'                               is set, each light will respond only to
corresponding switch position
'                               until circuit is reset.
'
' Assumptions
' 1. Times out after 10 seconds of all switches in the off position
'   regardless of whether all switches are learned yet
' 2. All switches must be off at start
'   (If switch 4 on when started, it starts up in audience-mode.)
' 3. All 4 switches must be switched on before that sequence is learned
' 4. Set bulb/LED outputs using bulb1, bulb2, bulb3 & bulb4
' 5. Set switch inputs using switch1, switch2, switch3 and switch4
' 6. first pattern is left to right, bulbs 1, 2, 3, 4
' 7. subsequent patterns are determined by last SWITCH turned off:
'   Switch 1 = 1234 order (bulb 1 first, then right in sequence)
```

```
'      Switch 2 = 2143 order (bulb 2 first, then left in sequence, wrapping
after first)
'      Switch 3 = 3412 order (bulb 3 first, then right in sequence,
wrapping after last)
'      Switch 4 = 4321 order (bulb 4 first, then left in sequence)
timeout = 750      'loop reset time approx 10 seconds
timeout_counter = 0
SetPin 11,2
SetPin 12,2
SetPin 13,2
SetPin 14,2
SetPin 15,8
SetPin 16,8
SetPin 17,8
SetPin 18,8
Pin(15) = 0:Pin(16)=0:Pin(17)=0:Pin(18)=0
If Pin(14) = 1 Then GoTo Audience_Lockdown
starting_lite = 1

' Start of program

do_reset:      ' reset position counter

If starting_lite = 1 Then
    position = 0      ' if starting with bulb 1, position reset to 0.
EndIf
If starting_lite = 2 Then
    position = 1      ' if starting with bulb 2, position reset to 1.
EndIf
If starting_lite = 3 Then
    position = 2      ' if starting with bulb 3, position reset to 2.
EndIf
If starting_lite = 4 Then
    position = 3      ' if starting with bulb 4, position reset to 3.
EndIf
flag0 = 0 : flag1 = 0 :flag2 = 0:flag3 = 0

' Learning loop

waiting_to_learn_loop:

If (Pin(11) = 1) And (flag0 = 0) Then GoTo learn0
If (Pin(12) = 1) And (flag1 = 0) Then GoTo learn1
If (Pin(13) = 1) And (flag2 = 0) Then GoTo learn2
If (Pin(14) = 1) And (flag3 = 0) Then GoTo learn3

' we have learnt that switch so light output accordingly

If flag0 = 1 Then
    If Pin(11) = 1 Then
```

```
        Pin(light0) = 1
    Else
        Pin(light0) = 0
    EndIf
EndIf
If flag1 = 1 Then
    If Pin(12) = 1 Then
        Pin(light1) = 1
    Else
        Pin(light1) = 0
    EndIf
EndIf
If flag2 = 1 Then
    If Pin(13) = 1 Then
        Pin(light2) = 1
    Else
        Pin(light2) = 0
    EndIf
EndIf
If flag3 = 1 Then
    If Pin(14) = 1 Then
        Pin(light3) = 1
    Else
        Pin(light3) = 0
    EndIf
EndIf

If (Pin(11) = 0) And (Pin(12) = 0) And (Pin(13) = 0) And (Pin(14) = 0) Then
    Pause 10
    timeout_counter = timeout_counter + 1
    If timeout_counter > timeout Then
        timeout_counter = 0
        GoTo do_reset
    EndIf
Else
    timeout_counter = 0
EndIf
GoTo waiting_to_learn_loop

' Learn a light position and set flag so we know that switch is done

learn0:

GoSub bulbset
flag0 = 1
light0 = bulb
GoTo learn_end

learn1:

GoSub bulbset
```

```
flag1 = 1
light1 = bulb
GoTo learn_end

learn2:

GoSub bulbset
flag2 = 1
light2 = bulb
GoTo learn_end

learn3:

GoSub bulbset
flag3 = 1
light3 = bulb
GoTo learn_end

learn_end:

If starting_lite = 1 Then      'if starting with 1st lamp, sequence =
1-2-3-4
    position = position + 1
    If position > 3 Then GoTo have_learnt_all
    GoTo waiting_to_learn_loop
EndIf
If starting_lite = 2 Then      'if starting with 2nd lamp, sequence =
2-1-4-3
    If position > 0 Then      'don't dec if position=0, will cause error
        position = position - 1
    Else
        position = 3
    EndIf
    If position = 1 Then GoTo have_learnt_all
    GoTo waiting_to_learn_loop
EndIf
If starting_lite = 3 Then      'if starting with 3rd lamp, sequence =
3-4-1-2
    position = position + 1
    If position > 3 Then
        position = 0
    EndIf
    If position = 2 Then GoTo have_learnt_all
    GoTo waiting_to_learn_loop
EndIf
If starting_lite = 4 Then      'if starting with 4th lamp, sequence =
4-3-2-1
    If position > 0 Then
        position = position - 1
    Else
```

```
        GoTo have_learnt_all
    EndIf
    GoTo waiting_to_learn_loop
EndIf

' now simply loop reacting to the switches
' timeout_counter value will increment every 10ms
' however if any light is on the timeout_counter is reset
' so this means the timeout will only
' occur after 10 seconds of all switches off

have_learnt_all:

If Pin(11) = 1 Then
    Pin(light0) = 1
    timeout_counter = 0
Else
    Pin(light0) = 0
EndIf
If Pin(12) = 1 Then
    Pin(light1) = 1
    timeout_counter = 0
Else
    Pin(light1) = 0
EndIf
If Pin(13) = 1 Then
    Pin(light2) = 1
    timeout_counter = 0
Else
    Pin(light2) = 0
EndIf
If Pin(14) = 1 Then
    Pin(light3) = 1
    timeout_counter = 0
Else
    Pin(light3) = 0
EndIf

If (flag0=1) And (flag1=1) And (flag2=1) And (flag3=1) Then
    all_flags = 1
EndIf
If (all_flags=1) And (Pin(11)=1) And (Pin(12)=0) And (Pin(13)=0) And
Pin(14)=0) Then
    starting_lite = 1
EndIf
If (all_flags=1) And (Pin(11)=0) And (Pin(12)=1) And (Pin(13)=0) And
Pin(14)=0) Then
    starting_lite = 2
EndIf
If (all_flags=1) And (Pin(11)=0) And (Pin(12)=0) And (Pin(13)=1) And
Pin(14)=0) Then
```

```
        starting_lite = 3
    EndIf
    If (all_flags=1) And (Pin(11)=0) And (Pin(12)=0) And (Pin(13)=0) And
    Pin(14)=1) Then
        starting_lite = 4
    EndIf
    Rem Print starting_lite;all_flags
    Pause 10
    timeout_counter = timeout_counter + 1
    If timeout_counter > timeout Then GoTo do_reset
    GoTo have_learnt_all

    bulbset:
    If position = 0 Then
        bulb = 15
    EndIf
    If position = 1 Then
        bulb = 16
    EndIf
    If position = 2 Then
        bulb = 17
    EndIf
    If position = 3 Then
        bulb = 18
    EndIf
    Rem Print position;bulb
    Return

    audience_lockdown:
    If Pin(11) = 1 Then
        Pin(15) = 1
    Else
        Pin(15) = 0
    EndIf
    If Pin(12) = 1 Then
        Pin(16) = 1
    Else
        Pin(16) = 0
    EndIf
    If Pin(13) = 1 Then
        Pin(17) = 1
    Else
        Pin(17) = 0
    EndIf
    If Pin(14) = 1 Then
        Pin(18) = 1
    Else
        Pin(18) = 0
    EndIf
    GoTo audience_lockdown
```

From:

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

Permanent link:

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

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

