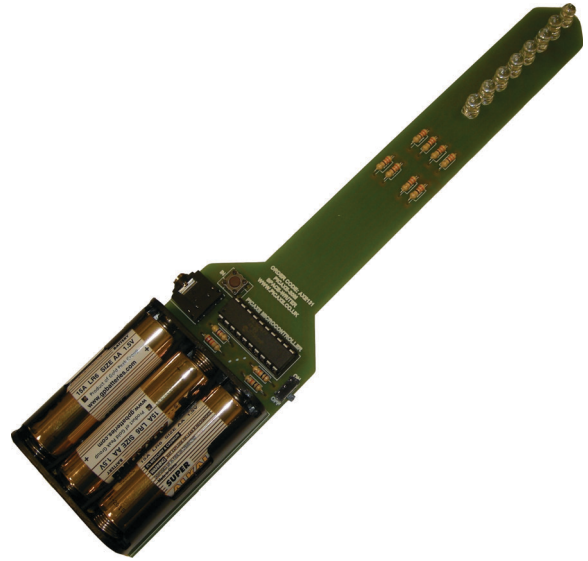


# PICAXE-20M SPACEWRITER KIT (AXE135)

---

## Contents:

- PCB AXE135 Project board PCB
- R1-8 330 resistor (orange orange brown gold)
- R9/10 10k resistor (brown black orange gold)
- R11 4k7 resistor (yellow violet red gold)
- R12 22k resistor (red red orange gold)
- LED1-8 LEDs
- SW1 On/Off Switch
- SW2 Tilt switch
- BB 4.5V Battery Box
- Socket 3.5mm Download socket
- IC1 20 pin IC socket
- IC1 PICAXE-20M



## Description:

The Spacewriter kit works by a principle called 'persistence of vision'. This principle means that a very short exposure of a light pattern to the eye creates a much longer reaction in the optic nerve in the eye. This means the light pattern appears to be 'seen' by the human eye for much longer than it was actually there. This is because sight is processed in the body by a chemical reaction in the optic nerves; this chemical reaction is slower than the rapid changes in light.

If you then apply lots of different light patterns very closely together, the eye can be 'tricked' into seeing larger patterns. With the Spacewriter kit, a single row of LEDs flicker on and off very quickly. If the 'wand' is then moved back and forth 'messages' can appear to be written in the air.

## Assembly Instructions:

1. Solder the resistors in position.
2. Solder the IC socket, download socket and on/off switch in position.
3. Solder the 8 LEDs in position, ensuring correct polarity (align flat edge with diagram on PCB).
4. Insert the tilt switch legs into the two holes in the rectangle directly above the word 'Switch'. Carefully fold over the tilt switch so that it lies flat on the PCB above the text 'PICAXE Microcontroller'. Solder in position, and use a glue gun to fix the tilt switch securely to the PCB.
5. Use a glue gun to stick the battery box securely to the PCB. Solder the two contacts in position.
6. Insert the PICAXE-20M microcontroller into the socket and connect 3xAA batteries.
7. Use the PICAXE software to develop a control program/flowchart, and then download the program to the board by connecting the PICAXE download cable (part AXE026 or AXE027).

## Tips for use:

To work effectively the Spacewriter must be flicked back and forth at a constant rate. This ensures the tilt switch activates the start of the message consistently. The LEDs will obviously be seen much better in a darkened room, but should also be visible during the day.

The best way to achieve a good light show is to hold the wand loosely near the end (half way down the battery box). Do not hold the wand too tightly, but allow it to flick backwards and forward in your hand, so the movement comes from the wrist and not the whole arm. Hold the wand out in front of you at hip height and flick backwards and forth rapidly by the wrist. With practise you should be able to get a good clear message displayed.

## Generating Each Letter

Have a look at this pattern (it might be helpful if you turn the paper on its side).  
The pattern is made up of 5 rows of 8 symbols.

```
--XXXXXX
-X--X---
-X--X---
--XXXXXX
-----
```

Hopefully you can see the letter 'A'. We can generate this pattern by having the 8 LEDs display each of the five lines in turn, only lighting the LEDs marked X.

If we then change this pattern into PICAXE BASIC code it will look like this:

```
let pins = %00111111
let pins = %01001000
let pins = %01001000
let pins = %00111111
let pins = %00000000
```

However as the PICAXE works at very high speed this pattern would process too fast, so we therefore insert some pauses between each line.

```
pause 2
let pins = %00111111
pause 2
let pins = %01001000
pause 2
let pins = %01001000
pause 2
let pins = %00111111
pause 2
let pins = %00000000
```

Finally, as the two centre lines are actually the same, we can re-write the code more efficiently like this:

```
letter_a:
  pause 2
  let pins = %00111111
  pause 2
  let pins = %01001000
  pause 4
  let pins = %00111111
  pause 2
  let pins = %00000000
  return
```

Adding a label 'letter\_a:' and the command 'return' at the end then allow the code to be used as a sub-procedure.

Appendix A contains a list of sub-procedures for all the different letters.

**PICAXE Program:**

The principle behind the PICAXE program is quite straight forward. When the tilt switch is activated a sequence of sub procedures are called. Each sub procedure displays one letter. By placing the sub procedures in different sequences different words can be spelt out.

You may need to adjust the 'pause 300' delay to make it easier to display the pattern when flicking the wand back and forth.

```

start:
    setfreq m8          ; fast operation
    pause 6
    let pins = %00000000
    return

main:
    ; wait for tilt switch
    if pin3 = 1 then do_message
    goto main

do_message:
    pause 300          ; adjust this delay
    gosub letter_p
    gosub letter_i
    gosub letter_c
    gosub letter_a
    gosub letter_x
    gosub letter_e
    goto main

letter_p:
    pause 2
    let pins = %01111111
    pause 2
    let pins = %01001000
    pause 2
    let pins = %01111000
    pause 2
    let pins = %00000000
    return

letter_i:
    pause 2
    let pins = %01000001
    pause 1
    let pins = %01111111
    pause 2
    let pins = %01000001
    pause 1
    let pins = %00000000
    return

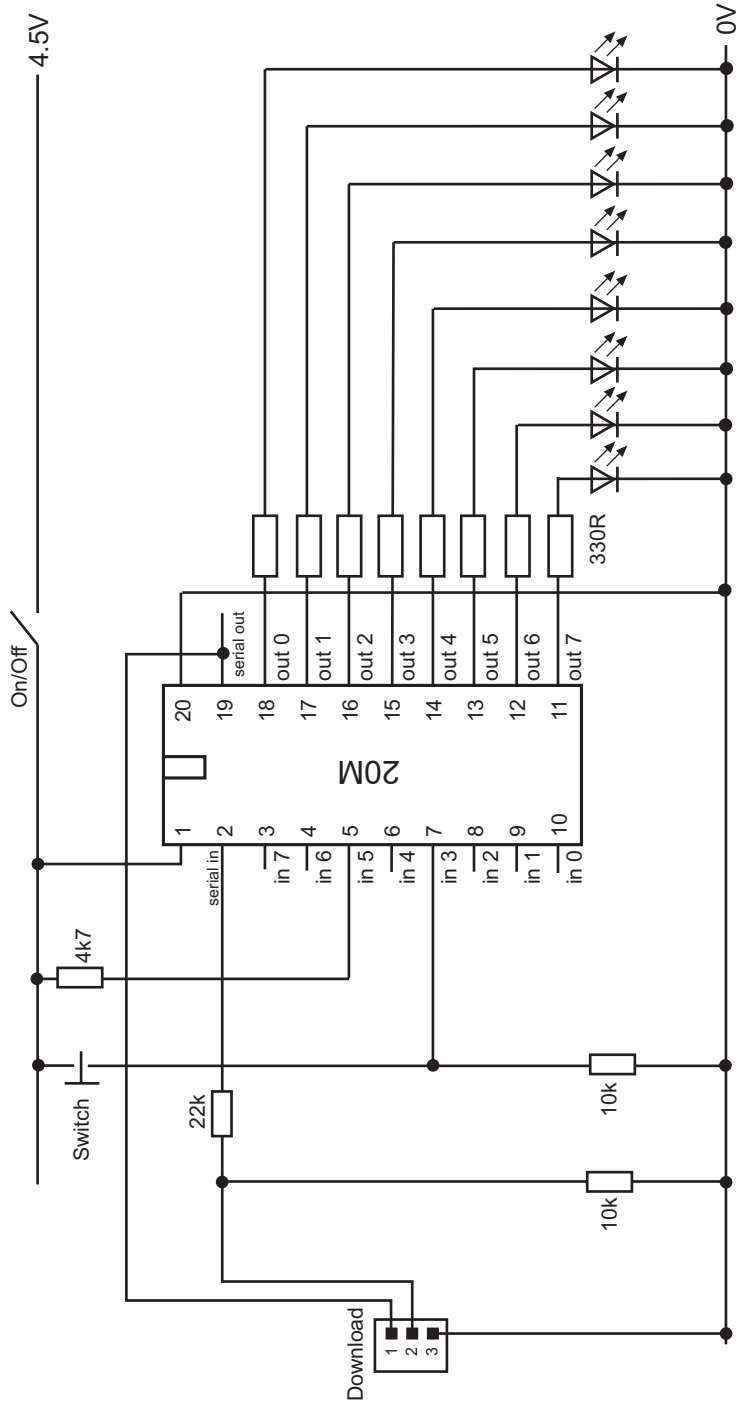
letter_c:
    pause 2
    let pins = %01111111
    pause 2
    let pins = %01000001

letter_a:
    pause 2
    let pins = %00111111
    pause 2
    let pins = %01001000
    pause 4
    let pins = %00111111
    pause 2
    let pins = %00000000
    return

letter_x:
    pause 2
    let pins = %01000001
    pause 1
    let pins = %00100010
    pause 1
    let pins = %00010100
    pause 1
    let pins = %00001000
    pause 1
    let pins = %00010100
    pause 1
    let pins = %00100010
    pause 1
    let pins = %01000001
    pause 1
    let pins = %00000000
    return

letter_e:
    pause 2
    let pins = %01111111
    pause 2
    let pins = %01001001
    pause 3
    let pins = %01000001
    pause 2
    let pins = %00000000
    return
    
```

### Circuit Diagram



## Appendix A

Note this appendix holds all the different letter sub-procedures. However there is not enough memory in the PICAXE-20M to hold all of the sub-procedures, so only cut and paste those that are required.

```

letter_a:
    pause 2
    let pins = %00111111
    pause 2
    let pins = %01001000
    pause 4
    let pins = %00111111
    pause 2
    let pins = %00000000
    return

letter_b:
    pause 2
    let pins = %00111110
    pause 2
    let pins = %01001001
    pause 4
    let pins = %01111111
    pause 2
    let pins = %00000000
    return

letter_c:
    pause 2
    let pins = %01111111
    pause 2
    let pins = %01000001
    pause 6
    let pins = %00000000
    return

letter_d:
    pause 2
    let pins = %01111111
    pause 2
    let pins = %01000001
    pause 3
    let pins = %00100010
    pause 2
    let pins = %00011100
    pause 1
    let pins = %00000000
    return

letter_e:
    pause 2
    let pins = %01111111
    pause 2
    let pins = %01001001
    pause 3
    let pins = %01000001
    pause 2
    let pins = %00000000
    return

letter_f:
    pause 2
    let pins = %01111111
    pause 2
    let pins = %01001000
    pause 4
    let pins = %01000000
    pause 1
    let pins = %00000000
    return

letter_g:
    pause 2
    let pins = %00111110
    pause 2
    let pins = %01000001
    pause 1
    let pins = %01001001
    pause 2
    let pins = %01001110
    pause 2
    let pins = %00000000
    return

letter_h:
    pause 2
    let pins = %01111111
    pause 1
    let pins = %00001000
    pause 4
    let pins = %01111111
    pause 1
    let pins = %00000000
    return

```

```

letter_i:
    pause 2
    let pins = %01000001
    pause 1
    let pins = %01111111
    pause 2
    let pins = %01000001
    pause 1
    let pins = %00000000
    return

letter_j:
    pause 2
    let pins = %01000000
    pause 2
    let pins = %01111111
    pause 2
    let pins = %01000001
    pause 2
    let pins = %01000010
    pause 1
    let pins = %00000000
    return

letter_k:
    pause 2
    let pins = %01111111
    pause 2
    let pins = %00001000
    pause 1
    let pins = %00010100
    pause 1
    let pins = %00100010
    pause 1
    let pins = %01000001
    pause 1
    let pins = %00000000
    return

letter_l:
    pause 2
    let pins = %00000001
    pause 4
    let pins = %01111111
    pause 2
    let pins = %00000000
    return

letter_m:
    pause 2
    let pins = %01111111
    pause 2
    let pins = %00100000
    pause 1
    let pins = %00010000
    pause 1
    let pins = %00001000
    pause 1
    let pins = %00000100
    pause 1
    let pins = %00000010
    pause 1
    let pins = %00000010
    pause 1
    let pins = %00001000
    pause 1
    let pins = %00010000
    pause 1
    let pins = %00100000
    pause 1
    let pins = %01111111
    pause 2
    let pins = %00000000
    return

letter_n:
    pause 2
    let pins = %01111111
    pause 2
    let pins = %00100000
    pause 1
    let pins = %00010000
    pause 1
    let pins = %00001000
    pause 1
    let pins = %00000100
    pause 1
    let pins = %01111110
    pause 2
    let pins = %00000000
    return

letter_o:
    pause 2
    let pins = %00111110
    pause 2
    let pins = %01000001
    pause 4
    let pins = %00111110
    pause 2
    let pins = %00000000
    return
    
```

```
letter_p:  pause 2
           let pins = %01111111
           pause 2
           let pins = %01001000
           pause 2
           let pins = %01111000
           pause 2
           let pins = %00000000
           return
```

```
letter_t:  pause 2
           let pins = %01000000
           pause 2
           let pins = %01111111
           pause 2
           let pins = %01000000
           pause 2
           let pins = %00000000
           return
```

```
letter_q:  pause 2
           let pins = %00111110
           pause 2
           let pins = %01000001
           pause 1
           let pins = %01000101
           pause 1
           let pins = %01000011
           pause 1
           let pins = %01111111
           pause 2
           let pins = %00000000
           return
```

```
letter_u:  pause 2
           let pins = %01111110
           pause 1
           let pins = %00000001
           pause 2
           let pins = %01111110
           pause 1
           let pins = %00000000
           return
```

```
letter_r:  pause 2
           let pins = %01111111
           pause 2
           let pins = %01001100
           pause 1
           let pins = %01001010
           pause 1
           let pins = %01001001
           pause 1
           let pins = %00000000
           return
```

```
letter_v:  pause 2
           let pins = %01100000
           pause 2
           let pins = %00010000
           pause 1
           let pins = %00001000
           pause 1
           let pins = %00000100
           pause 1
           let pins = %00000011
           pause 1
           let pins = %00000100
           pause 1
           let pins = %00000100
           pause 1
           let pins = %00000100
           pause 1
           let pins = %00001000
           pause 1
           let pins = %00010000
           pause 1
           let pins = %00100000
           pause 1
           let pins = %01111111
           pause 2
           let pins = %00000000
           return
```

```
letter_s:  pause 2
           let pins = %00111001
           pause 2
           let pins = %00010001
           pause 1
           let pins = %01001001
           pause 1
           let pins = %00100111
           pause 2
           let pins = %00000000
           return
```

```
letter_w:  pause 2
           let pins = %01111111
           pause 1
           let pins = %00000010
           pause 1
           let pins = %00000100
           pause 1
           let pins = %00001000
           pause 1
           let pins = %00000100
           pause 1
           let pins = %00000010
           pause 1
           let pins = %01111111
           pause 1
           let pins = %00000000
           return
```

```
letter_x:  pause 2
           let pins = %01000001
           pause 1
           let pins = %00100010
           pause 1
           let pins = %00010100
           pause 1
           let pins = %00001000
           pause 1
           let pins = %00010100
           pause 1
           let pins = %00100010
           pause 1
           let pins = %01000001
           pause 1
           let pins = %00000000
           return
```

```
letter_y:  pause 2
           let pins = %01000000
           pause 2
           let pins = %00100000
           pause 1
           let pins = %00011111
           pause 1
           let pins = %00100000
           pause 1
           let pins = %01000000
           pause 1
           let pins = %00000000
           return
```

```
letter_z:  pause 2
           let pins = %01000001
           pause 1
           let pins = %01100001
           pause 1
           let pins = %01010001
           pause 1
           let pins = %01001001
           pause 1
           let pins = %01000101
           pause 1
           let pins = %01000001
           pause 1
           let pins = %00000000
```

```
heart:     pause 4
           let pins = 56
           pause 2
           let pins = 124
           pause 2
           let pins = 126
           pause 2
           let pins = 63
           pause 2
           let pins = 126
           pause 2
           let pins = 124
           pause 2
           let pins = 56
           pause 2
           let pins = 0
           return
```