

Seven segment LEDs or multiple digit versions are used all over the place. Take a look at your washing machine, microwave, digital clock and so on. Ever wondered how they work. Well as ever with these things it's rarely as complicated as you might expect. Please go to immediate mode.

Copy the wiring image to the right on your breadboard. Notice the seven segment LED has a dot on the bottom left hand corner. This is actually a decimal point but it also shows us which way round to put it.

The **RED** wire goes from **3.3V** to the middle pin along the bottom. The **BLUE** wire goes from **DIGITAL** pin 2 to the first pin on the left along the bottom of the Seven Segment LED.

The **3.3V** is, inside the LED, connected to one end of each of the segments. To make one light up we therefore need to send a Ground signal to the segment.

**Please note** the middle top pin should not be connected. If you do you might reset the **computer** which, is not actually that bad, but you will have to wait for it to reboot. In immediate mode type in;

**PINMODE ( 2, 1 )**

**DIGITALWRITE ( 2, 0 )**

Notice when you enter the **DIGITALWRITE ( 2, 0 )** command the bottom left segment lights up. The 0 is the same as a ground signal so the segment circuit is completed and it lights up.

Unplug the **BLUE** wire from the **LED** and try connecting it to the other points (just not the one with a **RED X!**). You will see which point connects to which segment.

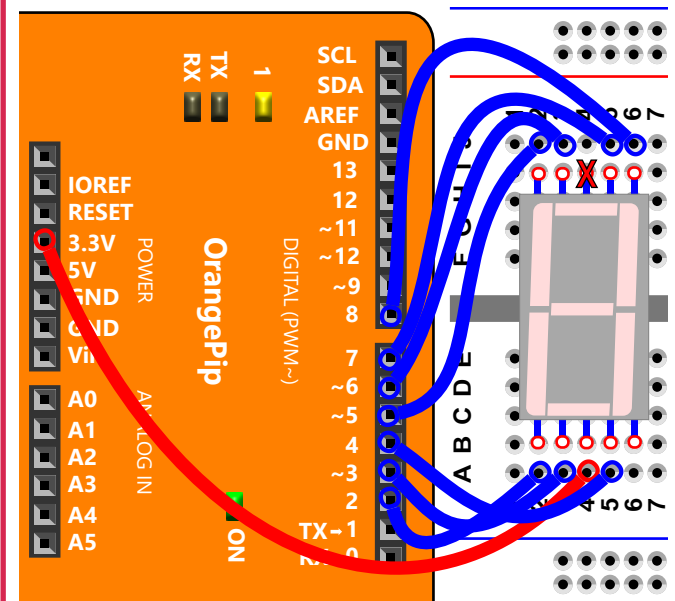
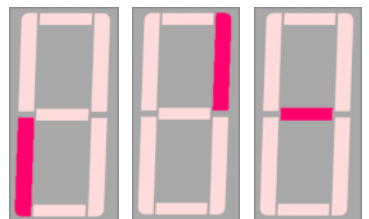
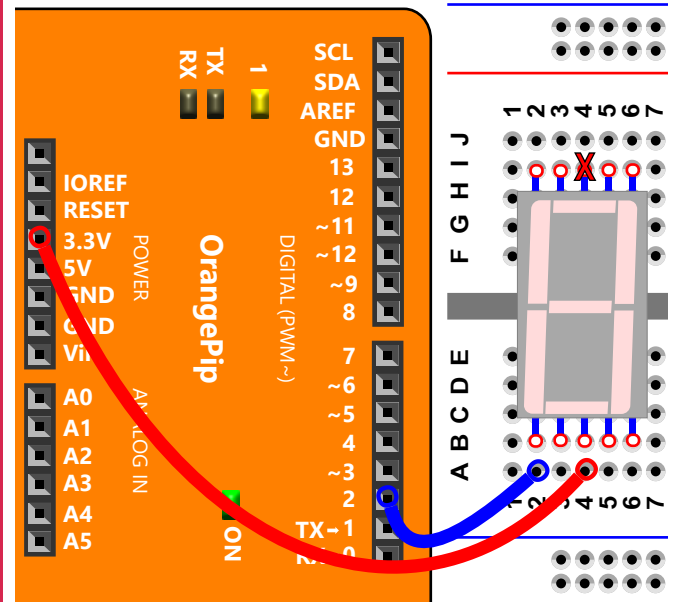
Now connect up the breadboard as displayed on the right and then go to the **Editor (F2)** and type in the program below.

```
FOR pin = 2 TO 8 LOOP
  PINMODE (pin , 1)
  REPEAT
  LOOP
  FOR pin = 2 TO 8 LOOP
    DIGITALWRITE (pin, 0)
  REPEAT
  WAIT (.3)
  FOR pin = 2 TO 8 LOOP
    DIGITALWRITE (pin, 1)
  REPEAT
  WAIT (.3)
  REPEAT
```

The first **LOOP** sets 7 **DIGITAL** pins to output mode. Then we run two **LOOPS**, one sending a low signal, turning the segments ON. The second sends a high signal switching them off.

### ADVANCED CHALLENGE:

See if you can make the LEDs snake around the display rather than all going on and off at the same time. Hint: Move the WAIT



In the **Editor** enter the program as displayed on the right;

As you can see we are directly controlling all of the segments individually. In this case a 0 is turning each segment on. When lit this looks like an 8.

Try changing some of the 0's to 1's and see if you can make other different numbers to be displayed. For example try 1, 0, 0, 0, 1, 0, 0. This will display the number 3 - clever eh!

It would be very messy to list all the **DIGITALWRITE** commands for the digits 0 to 9. What we need is some kind of matrix to store everything in so we can access it quickly and easily. To do this we use a special variable called an **ARRAY**.

```
FOR pin = 2 TO 9 LOOP
PINMODE ( pin, 1 )
REPEAT
LOOP
DIGITALWRITE ( 2, 0 )
DIGITALWRITE ( 3, 0 )
DIGITALWRITE ( 4, 0 )
DIGITALWRITE ( 5, 0 )
DIGITALWRITE ( 6, 0 )
DIGITALWRITE ( 7, 0 )
DIGITALWRITE ( 8, 0 )
REPEAT
```

**DIM LED (9, 6)** creates a new **ARRAY** variable indexed as shown here;

		SEGMENT						
		0	1	2	3	4	5	6
NUMBER	0	0	0	0	1	0	0	0
	1	1	1	0	1	1	1	0
	2	0	0	1	0	1	0	0
	3	1	0	0	0	1	0	0
	4	1	1	0	0	0	1	0
	5	1	0	0	0	0	0	1
	6	0	0	0	0	0	0	1
	7	1	1	0	1	1	0	0
	8	0	0	0	0	0	0	0
	9	1	0	0	0	0	0	0

In the **Editor** enter this program;

```
PROC setup
LOOP
INK = Raspberry
PRINTAT (0, 0); "Press 0-9"
key = INKEY - 48
IF key >= 0 AND key <= 9 THEN
FOR segment = 0 TO 6 LOOP
DIGITALWRITE (segment + 2, led(key, segment))
REPEAT
ENDIF
REPEAT
DEF PROC setup
FOR pin = 2 TO 8 LOOP
PINMODE (pin, 1)
REPEAT
FONTSIZE (10)
DIM led(9, 6)
FOR number = 0 TO 9 LOOP
FOR segment = 0 TO 6 LOOP
READ onoff
led(number, segment) = onoff
REPEAT
REPEAT
DATA 0, 0, 0, 1, 0, 0, 0
DATA 1, 1, 0, 1, 1, 1, 0
DATA 0, 0, 1, 0, 1, 0, 0
DATA 1, 0, 0, 0, 1, 0, 0
DATA 1, 1, 0, 0, 0, 1, 0
DATA 1, 0, 0, 0, 0, 0, 1
DATA 0, 0, 0, 0, 0, 0, 1
DATA 1, 1, 0, 1, 1, 0, 0
DATA 0, 0, 0, 0, 0, 0, 0
DATA 1, 0, 0, 0, 0, 0, 0
CLS
ENDPROC
```

This needs some explanation. The first line **PROC SETUP** jumps to a routine called **SETUP (DEF PROC SETUP)**. The sets the **DIGITAL** pins for output and the

The command **DIM LED (9, 6)** initialises a new **ARRAY** variable called **LED** and gives it an index of 0 to 9 down and 0 to 6 across. We use two **FOR LOOPS** to read a sequence of **DATA** values and store these in the **ARRAY** variable **LED**. In other words the variable **LED** now looks like this chart;

Now when we want to display a digit we just need to look up the arrangement of on's and off's in the **ARRAY** so **number 4** looks like "**1100010**" or "**off off on on on off on**" - remember in this case the off is sending a ground signal which switches the **segment** on.

In our program we're using a system variable that always has the code value of the current key being pressed stored in it. The variable, **INKEY** can be read to see if a key is pressed. The code value for the key **0** on your keyboard is **48**, **1** is **49**, **2** is **50** and so on. If we subtract **48** from the reading we're left with the values **0** to **9**. Next we check to make sure it is only **0** to **9** and **IF** so, display the key on screen and run the **LOOP** to set the segments on or off as defined in the **LED ARRAY** with the statement; **DIGITALWRITE (SEGMENT, LED (KEY, SEGMENT))**. The first use of **SEGMENT** here is being used for the **PIN** number also. Notice the use of the **ARRAY** variable **LED (KEY, SEGMENT)**.

**ADVANCED CHALLENGE:** This is your toughest challenge yet, but if you can do it then you have nailed **ARRAYS**. Given that the input key code for the letter 'a' is 97 could you modify the program to display letters instead of numbers. Hints: use 0 to 25 for the alphabet - define capital letters on the LED not lower case - don't forget to change the LED ARRAY to (25, 6) to support the full alphabet and increase the NUMBER LOOP to 25. Good luck and don't forget to change, and add, the DATA values from 10 digits to 26 alphabet characters. It's a lot of work but it is worth it!