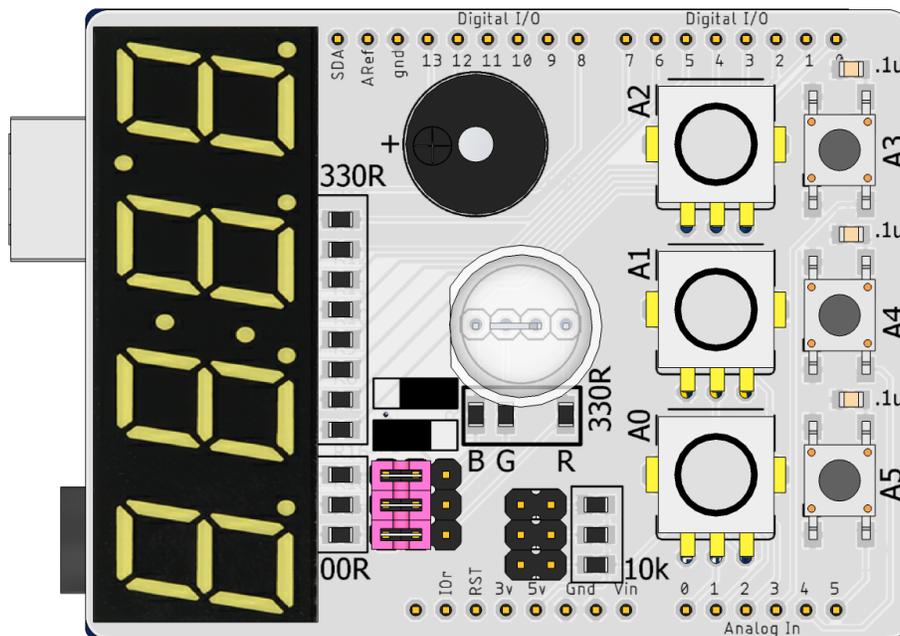


PART 1 (Changing over the board)

Because of the limited number of output pins available on the microcontroller, your board has jumpers to enable different functions. The jumpers are shown in pink below.

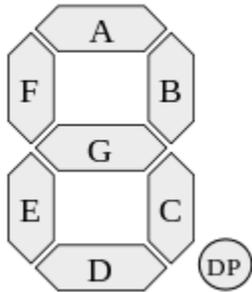


If the jumpers are to the left (as shown above) the upper 3 digits of the 4 digit display are enabled. If the jumpers are to the right, then the RGB LED is enabled. Most likely you will have to move the jumpers. Carefully pull up each of the three jumpers and move them to the right as shown.

Write a program which takes a global `int` variable call "val" (which must be in the range 0000-9999), and displays it in the 4 digit display. The "val" variable is to be set once in the program; however, I should be able to change val, rerun the program and have a different value displayed with no other changes to the program.

4 Digit Display Info

A 7-segment display is a matrix of leds than can be used to create any hex character (0-F). The segments are each labeled by a letter (A - G + dp for decimal point). The display on the dev board has a 4 character display on it.



The display on your board puts four of these into a single package:



When inserting this into the board (if not already done), the decimal points should be closest to the switches/pots on the board. Line up the pins carefully, so as not to bend the pins.

In the display, all of the segment a's are connected common, as are all the segment b's, and so on up to segment g's and dp's. These segments are connected to PORTD as follows:

Segment (common across digits)	PORTD Bit
A	d0
B	d1
C	d2
D	d3
E	d4
F	d5
G	d6
dp (decimal point)	d7

To control which digit is lit, a bit of PORTB is used as follows:

Common for character	PORTB Bit
Digit 0 (rightmost)	b0
Digit 1	b1
Digit 2	b2
Digit 3 (leftmost)	b3

For example, to display a 4 on Digit 2, I would need to turn on segments f, g, b, c. To do this, I would write a B01100110 to PORTD. I would also have to insure that bit2 of PORTB is set to 1.

```
// assuming PORTD (all bits) and PORTB (bit2) are set to outputs
PORTD = B01100110;
PORTB |= B100;      // or 1<<2
```

Now that your board is jumpered to enable all of the digits, each of the digits can be individually controlled.

PART 2: DELIVERABLE:

The TA will give you a number that `val` must be set to. Set the value, and demo the program. The `val` should be displayed on the 7 segment displays. A `val` like 543 should display as 0543

This must be checked off by TA, AND submitted to your M: drive.

Student Name: _____

TA Signature: _____

Date/Time _____

Checksum: _____

Things to remember/consider:

You need to break `val` down into individual digits. Use some combinations of `/10` and `%10` to do this.

Since the a's, b's, c's, etc., are connected common, it is impossible to display different values on the digits at the same time (i.e. it's not possible to display a 5 on digit 0, and a 7 on digit 1 at the same time). However, if you display a 5 on digit 0 for, say 1/60 of a second, and then display a 7 on digit 1 for 1/60 of a second, and repeat this rapidly, it will appear that 75 is displayed.

Remember to set all needed pins as outputs before trying to write a HIGH (or LOW) to any of these pins.

You are encouraged to develop some functions to help you do your program; it is up to you. Examples below; use other functions as you feel appropriate.

```
void ShowDig(uint8_t val, uint8_t pos); // this function sets character at pos to val
                                        // i.e. ShowDig(7,3) sets position 3 to 7
```