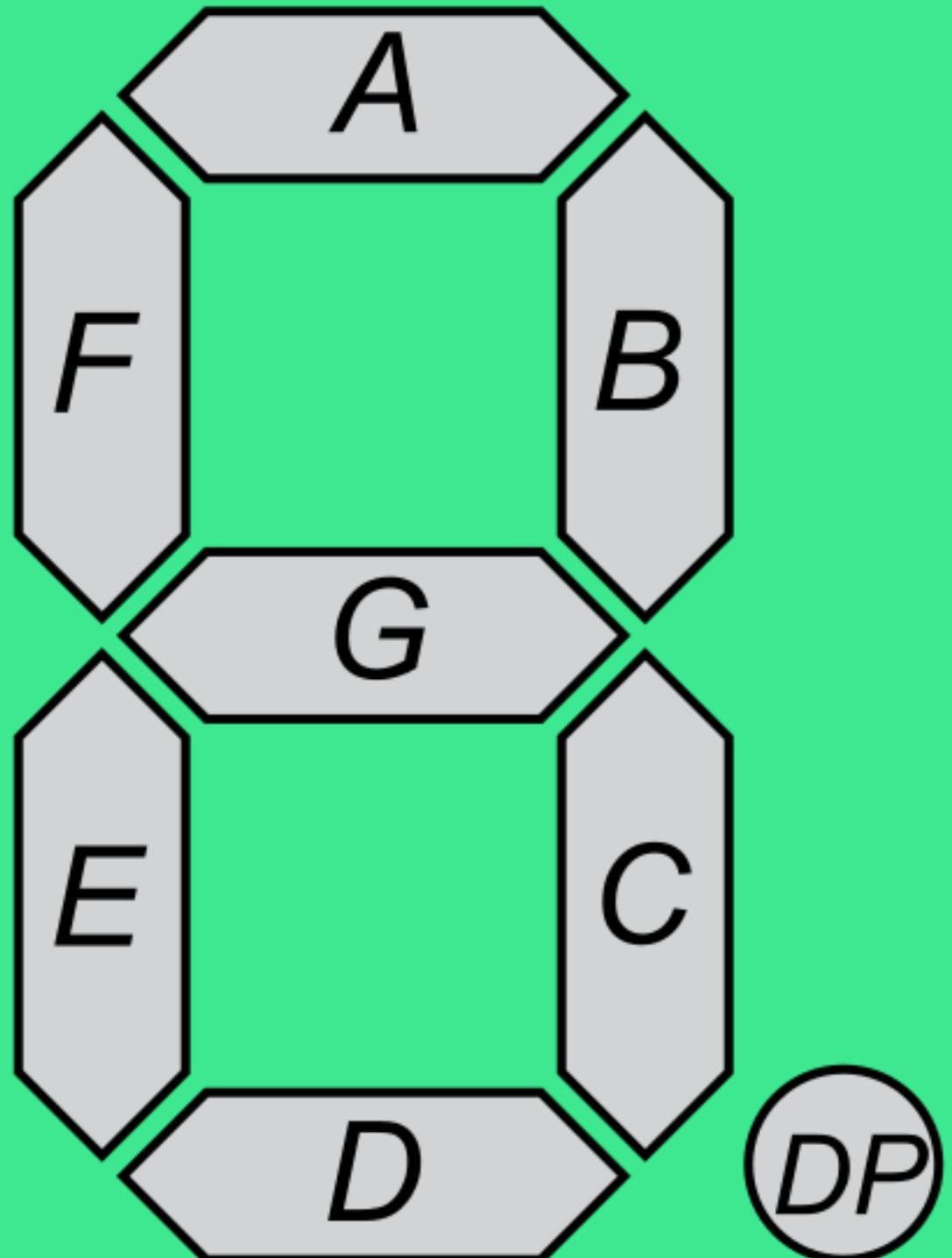


Introduction to Segment Display

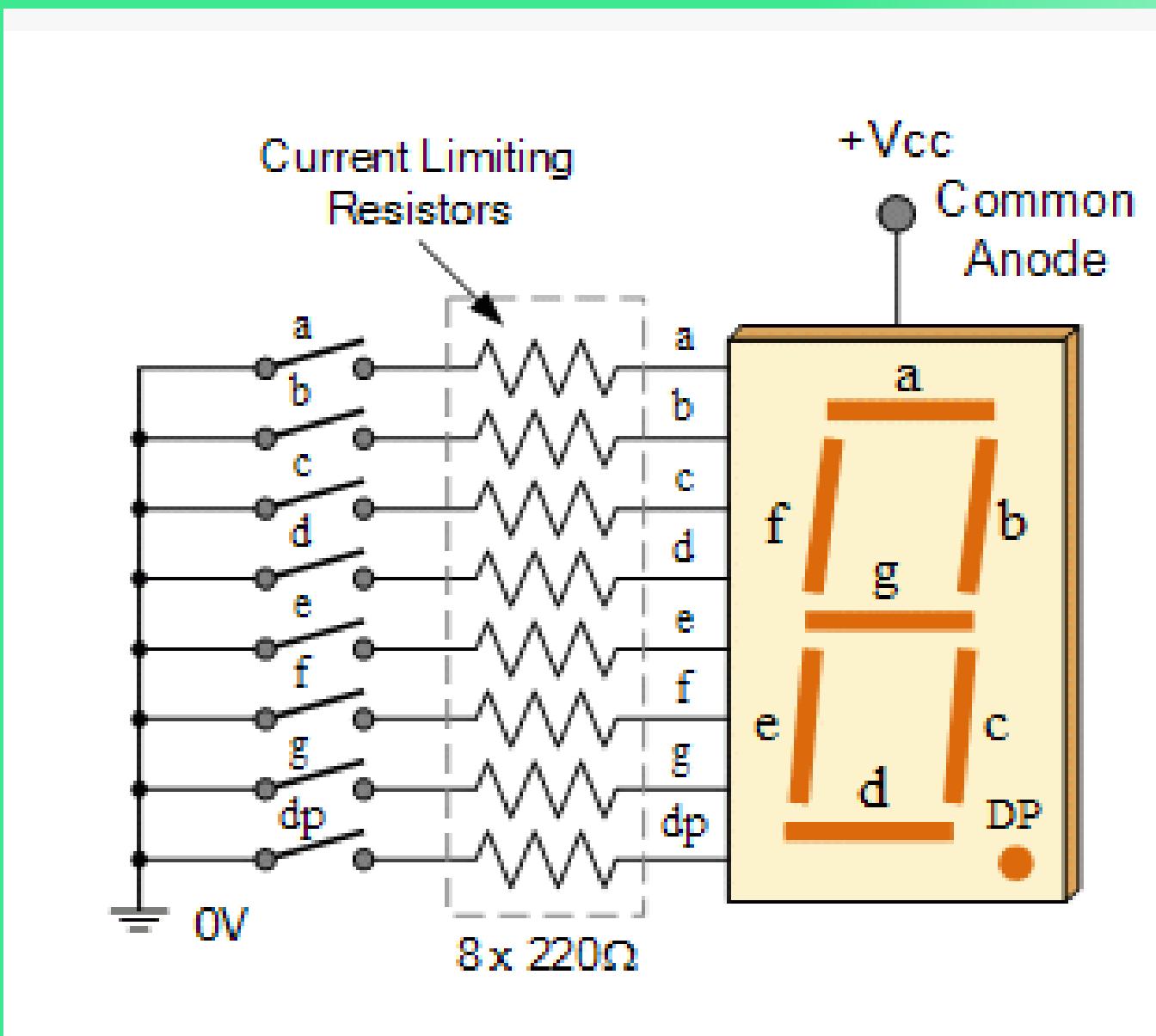
CSE 315
Peripherals & Interfacing
Rashik Rahman
Guest Lecturer



Agenda

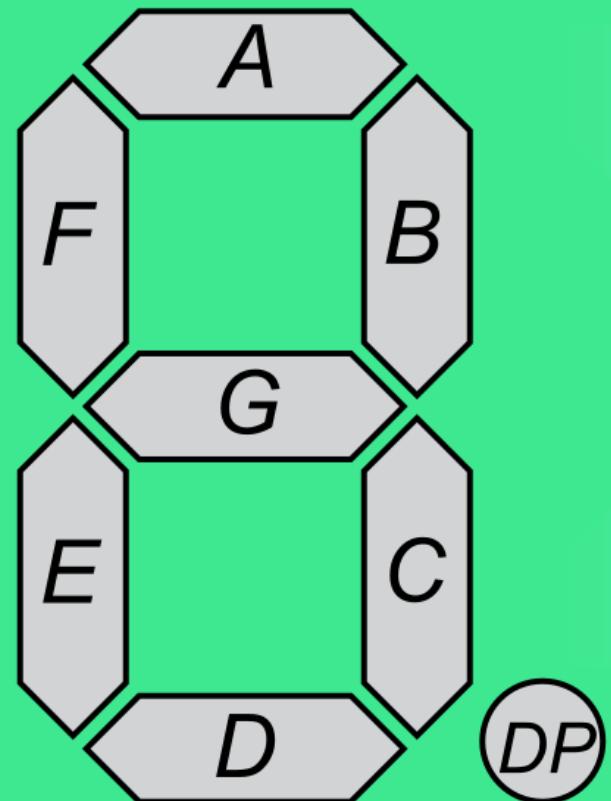
Today We'll learn about 7 segment display and it's arduino usage

What is 7 segment display?

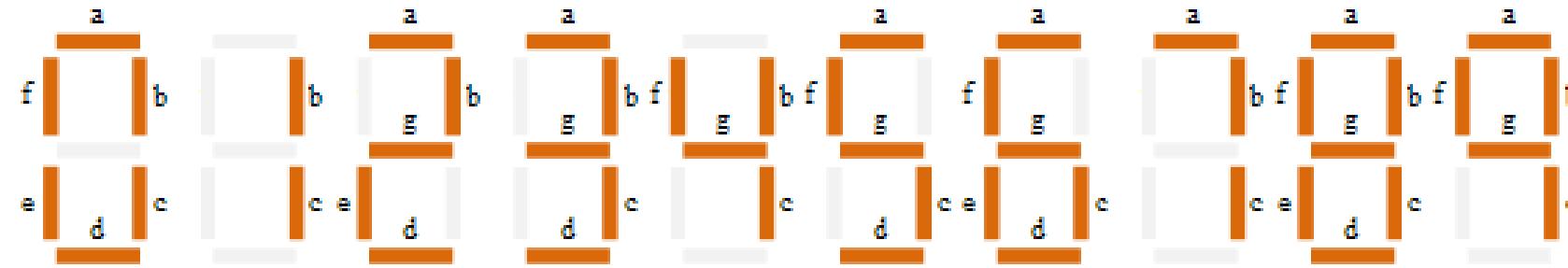


An LED or Light Emitting Diode, is a solid state optical pn-junction diode which emits light energy in the form of photons. The main advantage of light emitting diodes is that because of their small die size, several of them can be connected together within one small and compact package producing what is generally called a 7-segment Display.

Why 7 Segment?



7-Segment Display Segments for all Numbers.

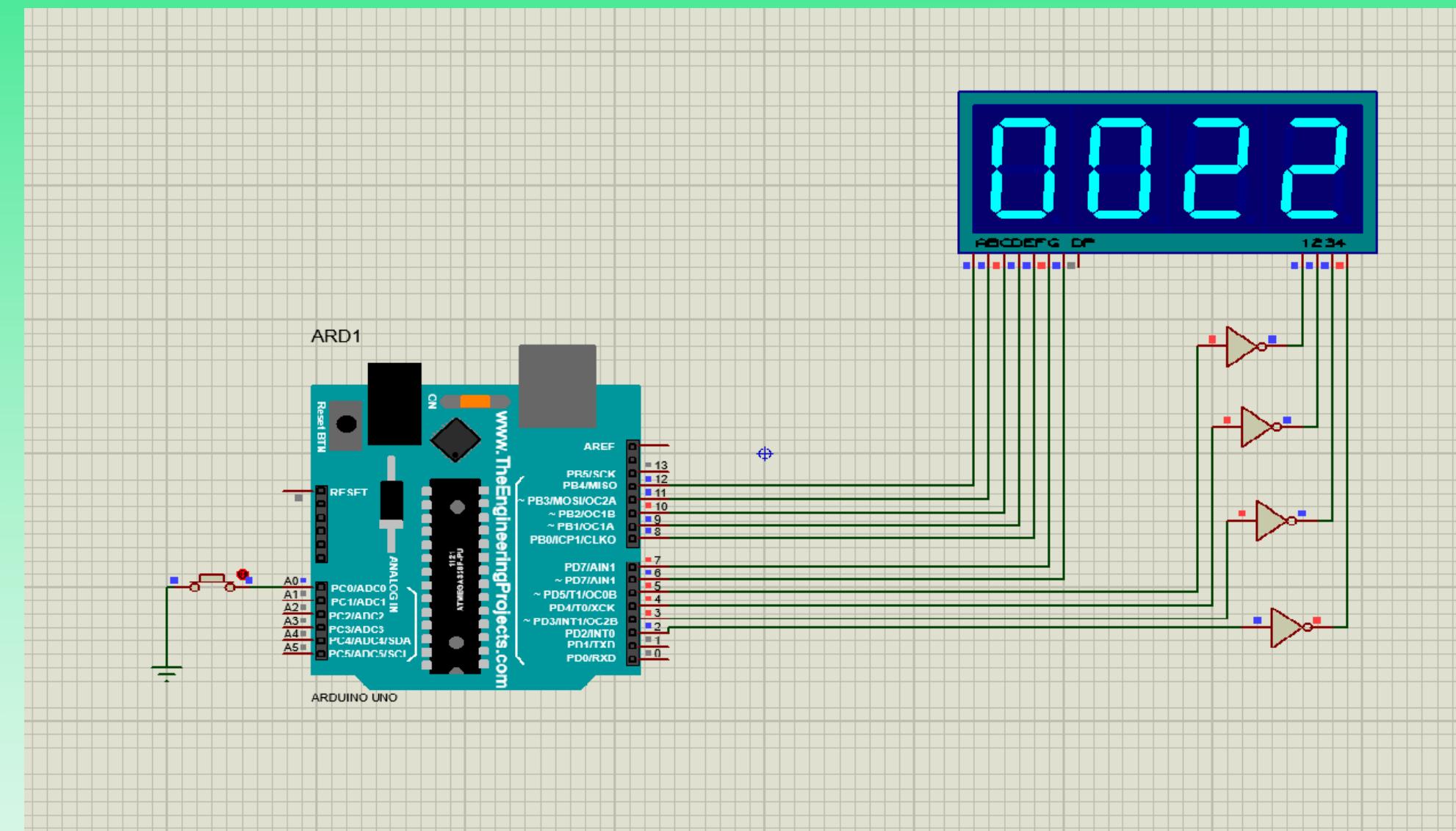


Then for a 7-segment display, we can produce a truth table giving the individual segments that need to be illuminated in order to produce the required decimal digit from 0 through 9 as shown below.

Truth Table of 7 segment

7-segment Display Truth Table

Decimal Digit	Individual Segments Illuminated						
	a	b	c	d	e	f	g
0	x	x	x	x	x	x	
1		x	x				
2	x	x		x	x		x
3	x	x	x	x			x
4		x	x			x	x
5	x		x	x		x	x
6	x		x	x	x	x	x
7	x	x	x				
8	x	x	x	x	x	x	x
9	x	x	x			x	x



Here "x" means switch is closed/connected so that current can pass through that pin

Some Use Case



Counter



Speed measuring device



Digital Clock



Cashier machine



Digital meter



Microwave oven



Washing machine

etc.

Lets see codes &Simulation

```

// counter button definition
#define button A0

// segment pin definitions
#define SegA 12
#define SegB 11
#define SegC 10
#define SegD 9
#define SegE 8
#define SegF 7
#define SegG 6
// common pins of the four digits definitions
#define Dig1 5
#define Dig2 4
#define Dig3 3
#define Dig4 2

// variable declarations
byte current_digit;
int count = 0;

//What is an interrupt?
//The program running on a controller is normally running sequentially instruction by instruction.
//An interrupt is an external event that interrupts the running program and runs a special interrupt service routine (ISR).
//After the ISR has been finished, the running program is continued with the next instruction. Instruction means a single machine instruction, not a line of C or C++ code.

ISR(TIMER1_OVF_vect) // Timer1 interrupt service routine (ISR)
{
  disp_off(); // turn off the display

  switch (current_digit)
  {
    case 1:
      disp(count / 1000); // prepare to display digit 1 (most left)
      digitalWrite(Dig1, LOW); // turn on digit 1
      break;

    case 2:
      disp( (count / 100) % 10); // prepare to display digit 2
      digitalWrite(Dig2, LOW); // turn on digit 2
      break;

    case 3:
      disp( (count / 10) % 10); // prepare to display digit 3
      digitalWrite(Dig3, LOW); // turn on digit 3
      break;

    case 4:
      disp(count % 10); // prepare to display digit 4 (most right)
      digitalWrite(Dig4, LOW); // turn on digit 4
  }

  current_digit = (current_digit % 4) + 1;
}

void setup()
{
  pinMode(button, INPUT_PULLUP);
  pinMode(SegA, OUTPUT);
  pinMode(SegB, OUTPUT);
  pinMode(SegC, OUTPUT);
  pinMode(SegD, OUTPUT);
  pinMode(SegE, OUTPUT);
  pinMode(SegF, OUTPUT);
  pinMode(SegG, OUTPUT);
  pinMode(Dig1, OUTPUT);
  pinMode(Dig2, OUTPUT);
  pinMode(Dig3, OUTPUT);
  pinMode(Dig4, OUTPUT);

  disp_off(); // turn off the display

  // Timer1 module overflow interrupt configuration
  TCCR1A = 0;
  TCCR1B = 1; // enable Timer1 with prescaler = 1 ( 16 ticks each 1 µs)
  TCNT1 = 0; // set Timer1 preload value to 0 (reset)
  TIMSK1 = 1; // enable Timer1 overflow interrupt
}

```

```

// main loop
void loop()
{
  if(digitalRead(button) == 0)
  {
    // count++; // increment 'count' by 1
    // if(count == 9999)
    //   count = 0;
    count=9999;
    delay(200); // wait 200 milliseconds
  }
}

```

```
void disp(byte number)
{
    switch (number)
    {
        case 0: // print 0
            digitalWrite(SegA, LOW);
            digitalWrite(SegB, LOW);
            digitalWrite(SegC, LOW);
            digitalWrite(SegD, LOW);
            digitalWrite(SegE, LOW);
            digitalWrite(SegF, LOW);
            digitalWrite(SegG, HIGH);
            break;

        case 1: // print 1
            digitalWrite(SegA, HIGH);
            digitalWrite(SegB, LOW);
            digitalWrite(SegC, LOW);
            digitalWrite(SegD, HIGH);
            digitalWrite(SegE, HIGH);
            digitalWrite(SegF, HIGH);
            digitalWrite(SegG, HIGH);
            break;

        case 2: // print 2
            digitalWrite(SegA, LOW);
            digitalWrite(SegB, LOW);
            digitalWrite(SegC, HIGH);
            digitalWrite(SegD, LOW);
            digitalWrite(SegE, LOW);
            digitalWrite(SegF, HIGH);
            digitalWrite(SegG, LOW);
            break;

        case 3: // print 3
            digitalWrite(SegA, LOW);
            digitalWrite(SegB, LOW);
            digitalWrite(SegC, LOW);
            digitalWrite(SegD, LOW);
            digitalWrite(SegE, HIGH);
            digitalWrite(SegF, HIGH);
            digitalWrite(SegG, LOW);
            break;

        case 4: // print 4
            digitalWrite(SegA, HIGH);
            digitalWrite(SegB, LOW);
            digitalWrite(SegC, LOW);
            digitalWrite(SegD, HIGH);
            digitalWrite(SegE, HIGH);
            digitalWrite(SegF, LOW);
            digitalWrite(SegG, LOW);
            break;

        case 5: // print 5
            digitalWrite(SegA, LOW);
            digitalWrite(SegB, HIGH);
            digitalWrite(SegC, LOW);
            digitalWrite(SegD, LOW);
            digitalWrite(SegE, HIGH);
            digitalWrite(SegF, LOW);
            digitalWrite(SegG, LOW);
            break;

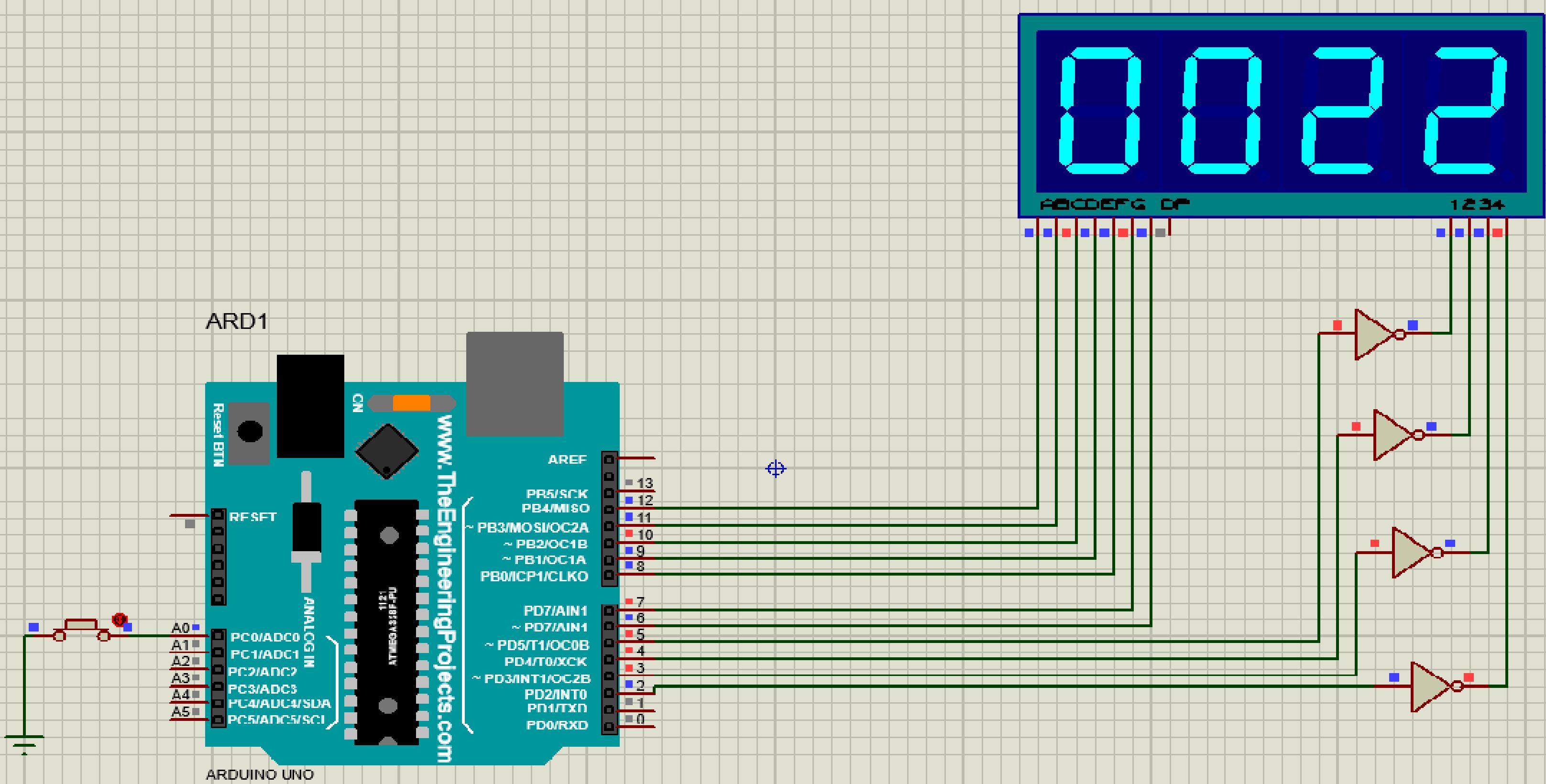
        case 6: // print 6
            digitalWrite(SegA, LOW);
            digitalWrite(SegB, HIGH);
            digitalWrite(SegC, LOW);
            digitalWrite(SegD, LOW);
            digitalWrite(SegE, LOW);
            digitalWrite(SegF, LOW);
            digitalWrite(SegG, LOW);
            break;

        case 7: // print 7
            digitalWrite(SegA, LOW);
            digitalWrite(SegB, LOW);
            digitalWrite(SegC, LOW);
            digitalWrite(SegD, HIGH);
            digitalWrite(SegE, HIGH);
            digitalWrite(SegF, HIGH);
            digitalWrite(SegG, HIGH);
            break;

        case 8: // print 8
            digitalWrite(SegA, LOW);
            digitalWrite(SegB, LOW);
            digitalWrite(SegC, LOW);
            digitalWrite(SegD, LOW);
            digitalWrite(SegE, LOW);
            digitalWrite(SegF, LOW);
            digitalWrite(SegG, LOW);
            break;

        case 9: // print 9
            digitalWrite(SegA, LOW);
            digitalWrite(SegB, LOW);
            digitalWrite(SegC, LOW);
            digitalWrite(SegD, LOW);
            digitalWrite(SegE, HIGH);
            digitalWrite(SegF, LOW);
            digitalWrite(SegG, LOW);
            break;
    }
}

void disp_off()
{
    digitalWrite(Dig1, HIGH);
    digitalWrite(Dig2, HIGH);
    digitalWrite(Dig3, HIGH);
    digitalWrite(Dig4, HIGH);
}
```



QUIZ?

- What's a 7 segment display?
- Write the orientation of 7 segments for 5.
- Is there any need of 7 segment display in 2020?
Give proper reasoning for your answer.

Thank you!