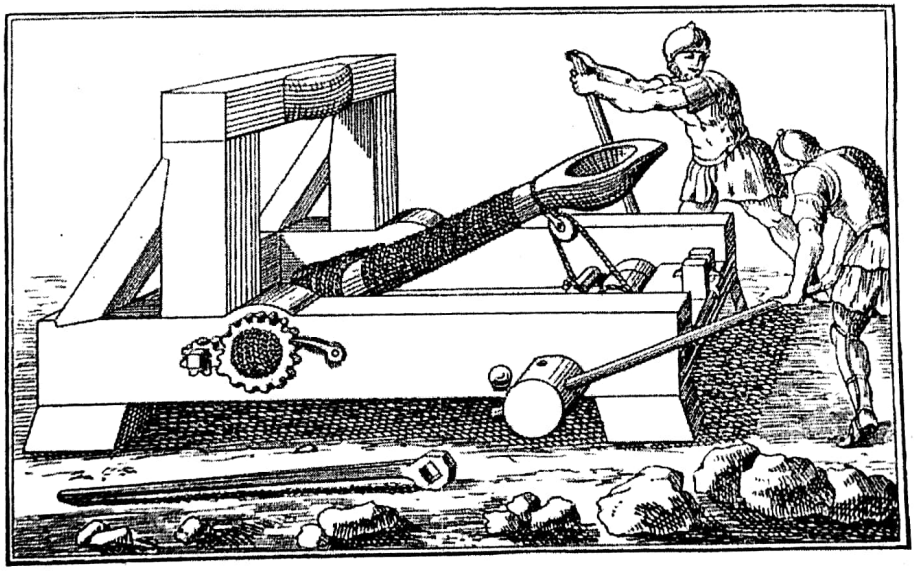
DEPARTMENT
OF
ELECTRONICS AND COMMUNICATION ENGINEERING

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Handout/Assignment-1
for
Engineering Design Project-I (UTA013)



NAME OF STUDENT: DIVYA AGRAWAL **INSTRUCTOR**
Roll No. 101715043
Group: ENC-2

ASSIGNMENT - 1
INPUT / OUTPUT INTERFACE DESIGN

Exercise 1 – Blink

To blink the LED with an Arduino for 1 sec on time and 2 seconds off time.

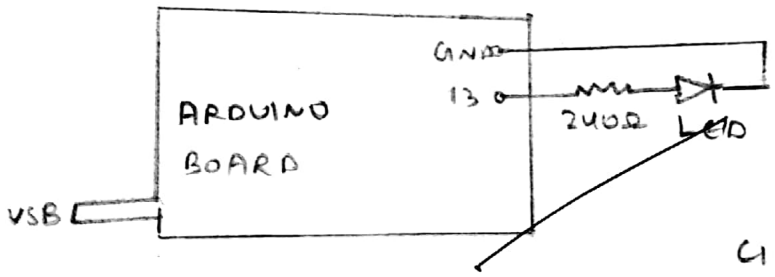
Hardware Required

- Arduino Board
- LED and the Resistor

Circuit description

To build the circuit, attach a 440-ohm resistor to either leg of the diode. Attach the leg of the LED connected to the flat edge of the body (the negative leg, called the cathode) to ground. Connect the remaining leg (the positive leg, called the anode) to pin 13. Then plug your Arduino board into your computer, start the Arduino program, and upload the code.

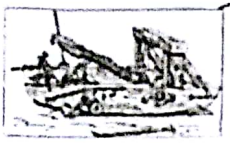
Code



CIRCUIT DIAGRAM

```
void setup()
{
// put setup code once to run here:
pinMode (13, OUTPUT); // to enter pin no. and mode in
                         paranthesis

}
void loop
{
// put your main code here:
digitalWrite (13, HIGH); // enter pin no. and mode as in
                            HIGH OR LOW WITHIN
                            PARANTHESIS ..
```



```
delay (2000);  // switches on LED for 2000 milliseconds.
digitalWrite (13, LOW);  // switches off LED at Pin 13 is off.
delay (2000);  // switches LED off for 2000 milliseconds.
}
```

Schematic

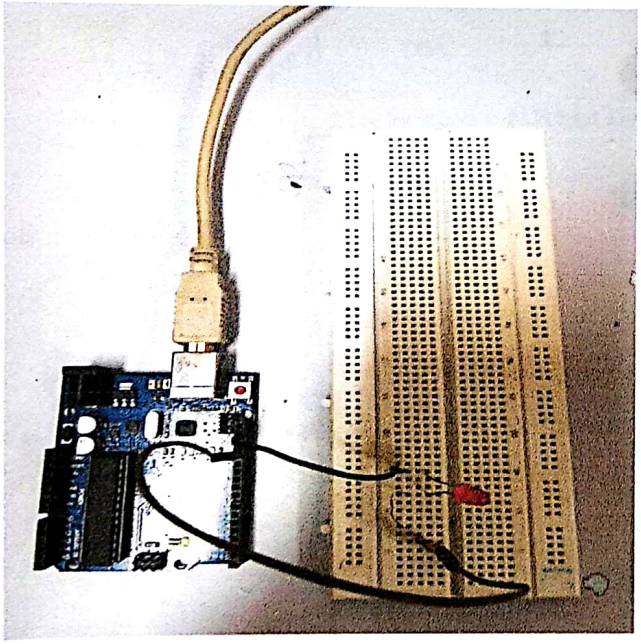


Fig.1

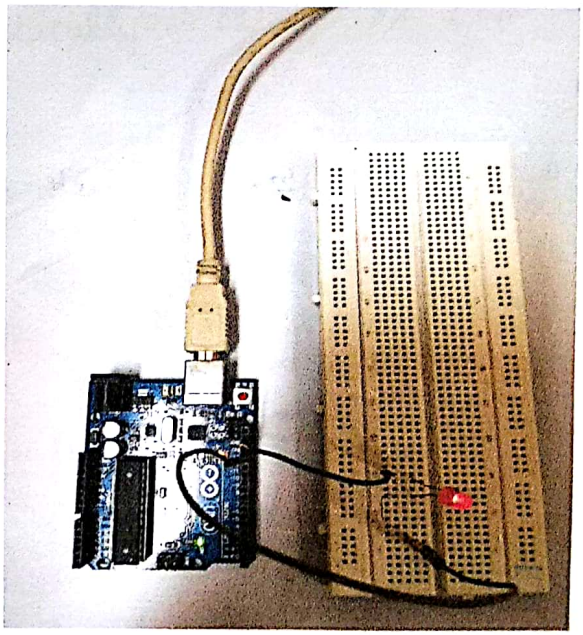


Fig.2

Figure 1 shows that LED is OFF for 2 seconds

Figure 2 shows that LED is ON for 2 seconds

Reflections:

* Through this experiment, coading through arduino setup was enhanced
* It helped me to practically blink an LED device using Arduino
* Various applications of arduino can be performed.

Exercise 2 – PushButton

Connect Pushbutton to turn on the built-in LED on pin 13 i.e.when you press the button LED should glow otherwise it should be in off state.

Hardware

- Arduino Board
- momentary button or switch
- 10K ohm resistor
- breadboard
- hook-up wire

Circuit description

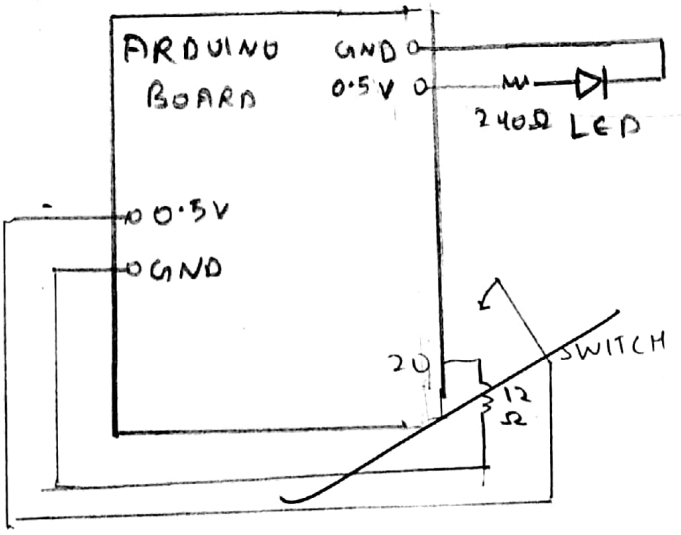
Connect three wires to the Arduino board. The first two, red and black, connect to the two long vertical rows on the side of the breadboard to provide access to the 5 volt supply and ground. The third wire goes from digital pin 10 to one leg of the pushbutton. The same leg of the button connects through a pull-down resistor (here 10 KOhms) to ground. The other leg of the button connects to the 5 volt supply.

When the pushbutton is open (unpressed) there is no connection between the two legs of the pushbutton, so the pin is connected to ground (through the pull-down resistor) and we read a LOW. When the button is closed (pressed), it makes a connection between its two legs, connecting the pin to 5 volts, so that we read a HIGH.

You can also wire this circuit the opposite way, with a pullup resistor keeping the input HIGH, and going LOW when the button is pressed. If so, the behaviour of the sketch will be reversed, with the LED normally on and turning off when you press the button.

If you disconnect the digital I/O pin from everything, the LED may blink erratically. This is because the input is "floating" - that is, it will randomly return either HIGH or LOW. That's why you need a pull-up or pull-down resistor in the circuit.

Code



CIRCUIT DIAGRAM.

```
void setup ( )
{ // put your setup code here, to run once :
  pinMode (13 , OUTPUT); // Assign pin no. to get output
  pinMode ( 13, INPUT); // Assign the pin no. to get input.
}
void loop() // put your main code here to run repeatedly.
{ if (digitalRead == HIGH) // condition for switch to be On.
  digital write (13, HIGH); // operation to be performed when switch is ON.
    else
  digital write (13, LOW); // when switch is OFF, LED turns off)
}.
```

Schematic

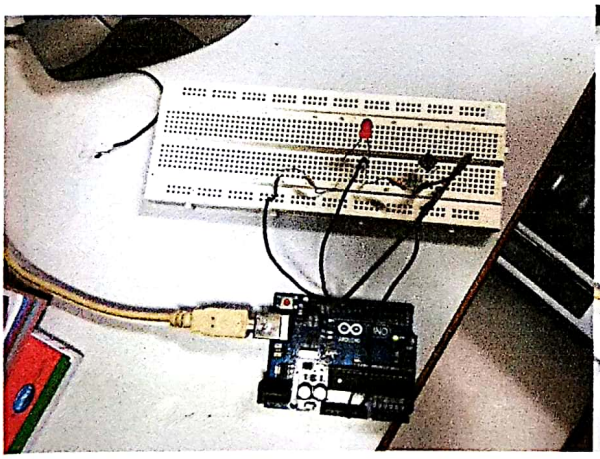


Fig 1

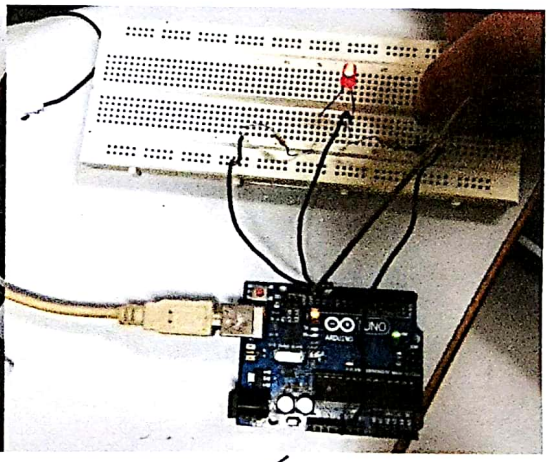


Fig 2

Fig. 1 shows when push button is unpressed, LED is turned

%

Fig. 2 shows that on pressing the push button, LED is turned ON.

Reflections:

* This experiment enlightened me with the knowledge about how hardware and software are linked (Arduino here)
* We understood the usage of push button and its operation on arduino board linked with the blinking of LED.
* Coding and technological skills were enhanced.

Assignment Tasks:

A. Change the amount of time the LED is off to 2 second. (Leaving the amount of time the LED is on at 1 second.)
B. Change the pin to which the LED is connected from pin 13 to pin 2.
C. Hook up 8 LEDs to pins 2 through 9 (with resistors, of course.) Modify the code to turn on each one in order and then extinguish them in order.
(HINT: hook them up one additional LED at a time and make sure the new one works before you add the next one.)

ASSIGNMENT TASK-B:

CODE:

```
void setup ()     // put your setup code here,
                     to move once
{
  pinMode (2, OUTPUT); // sets pin 2 of Arduino
                          as output pin
}
  void loop() // Main loop is written here
                 and it runs repeatedly.
{
digitalWrite (2, HIGH); // sets output at pin 2
                           as high.
delay (1000); // output remains same for 1 sec.
digitalWrite (2, LOW); // sets output pin 2 as LOW.
delay (2000); // current output remains same
}                for 2 sec.
```

ASSIGNMENT A

```
void setup() // put your setup code here, to run once
{
  pinMode (13, OUTPUT); // set pin 3 of Arduino as output pin.
}
void loop() // main loop is written here and it runs repeatedly.
{
  digitalWrite (13, HIGH); // sets output at pin 13 as high.
  delay (1000);
  digitalWrite (13, LOW); // sets output pin13 as low
  delay (2000); // delays time for 2000 millisec.
}
```

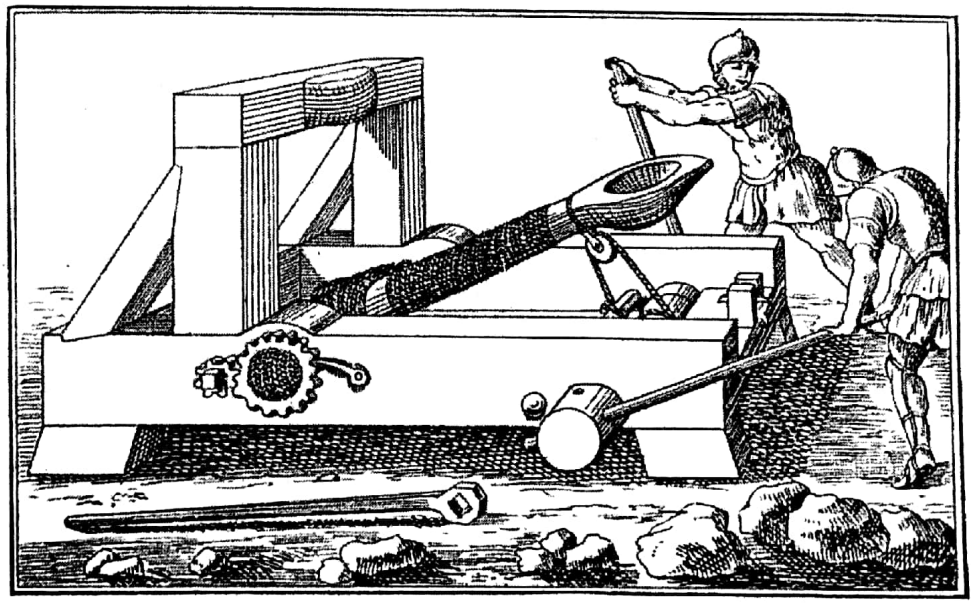
ASSIGNMENT TASK-1

```
void setup() // setup code is written here, it runs once.
{ for (int thisPin=2; thisPin<=9; thisPin++)
    // loop to initialize each pin as output.
  { pinMode(thisPin, OUTPUT); }
    // sets pins as OUTPUT pin.
}
void loop() // main code is written to run repeatedly
{ for (int thisPin=2; thisPin<=9; thisPin++)
    // loop to turn Pin ON.
  { digitalWrite (thisPin, HIGH);
    delay(1000); // each pin turns on after 1sec.
  }
  for (int thisPin=2; thisPin<=9; thisPin++)
    // loop to turn pin off
  { digitalWrite (thisPin, LOW); // each pin turns off
    delay(1000);                 after 1sec.
  }
```

DEPARTMENT
OF
ELECTRONICS AND COMMUNICATION ENGINEERING

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Handout/Assignment-2
for
Engineering Design Project-I (UTA013)



NAME OF STUDENT: DIVYA AGRAWAL
Roll No. 1017 1043
Group: ENC-2

INSTRUCTOR

ASSIGNMENT - 2
Study of ICs

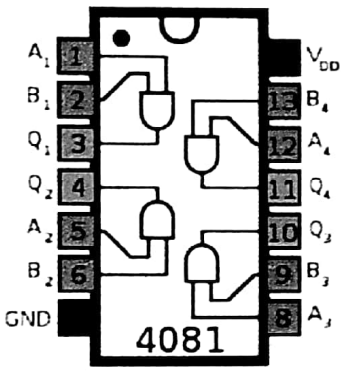
Exercise 1 – To verify the function tables of CD4027 and CD4081 ICs.

Hardware Required

Digital Trainer Kit
CD 4027 and CD4081
Single core connecting wires

Theory

The data sheet of CD4027 and CD4081 is given below.

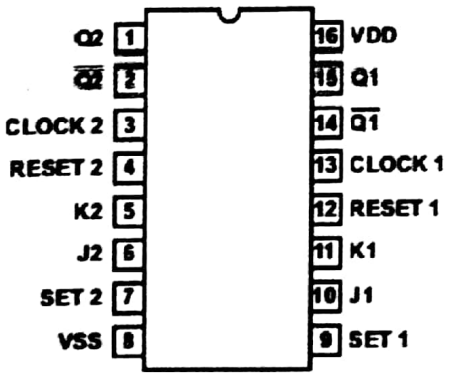


(a)

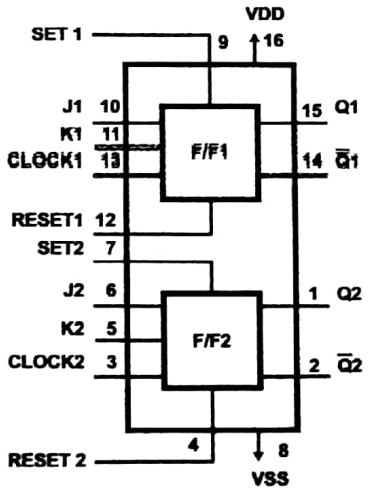
Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

(b)

Figure 1: Datasheet of CD4081 IC (a) pin diagram (b) functional truth table



(a)



(b)

Figure 2: CD4027 IC (a) pin diagram (b) Internal architecture

ASSIGNMENT - 2
Study of ICs

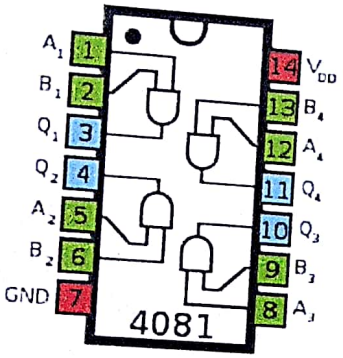
Exercise 1 – To verify the function tables of CD4027 and CD4081 ICs.

Hardware Required

Digital Trainer Kit
CD 4027 and CD4081
Single core connecting wires

Theory

The data sheet of CD4027 and CD4081 is given below.

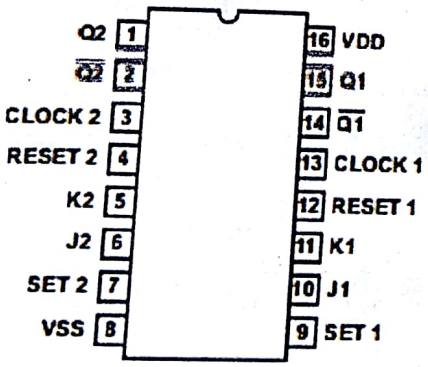


(a)

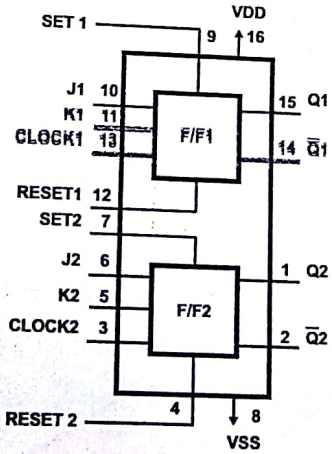
Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

(b)

Figure 1: Datasheet of CD4081 IC (a) pin diagram (b) functional truth table



(a)



(b)

Figure 2: CD4027 IC (a) pin diagram (b) Internal architecture

Trigger	Inputs		Output				Inference
			Present State		Next State		
CLK	J	K	Q	Q'	Q	Q'	
⊠	x	x	-		-		Latched
⊓	0	0	0	1	0	1	No Change
⊓			1	0	1	0	
⊓	0	1	0	1	0	1	Reset
⊓			1	0	0	1	
⊓	1	0	0	1	1	0	Set
⊓			1	0	1	0	
⊓	1	1	0	1	1	0	Toggles
⊓			1	0	0	1	

Schematic

CD 4027 IC

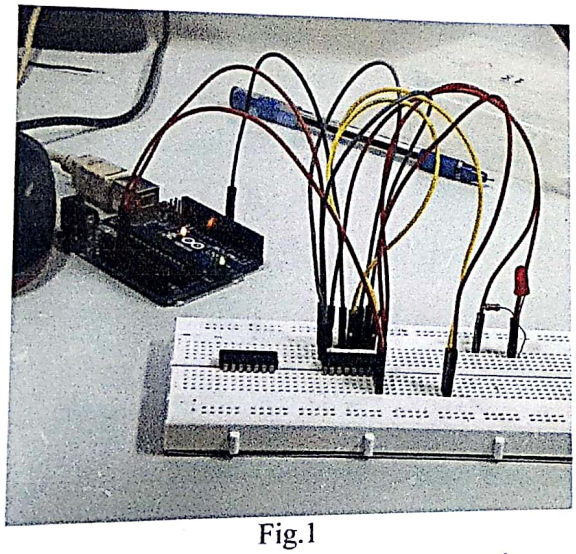


Fig.1

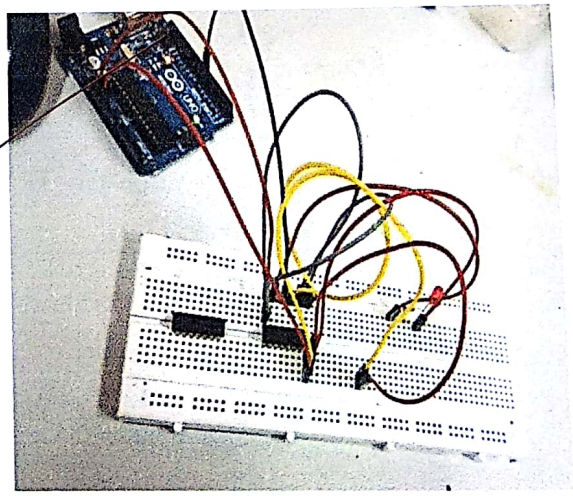


Fig.2

The figure 1 shows the working of JK flip flop.

In Fig.2, LED is ON.

CD 4027 is JK flip flop that is used for data storing. Two similar or equal JK flip flop are contained in IC.

Fig. 2 & 1 shows the case of toggling.

CD 4081 IC

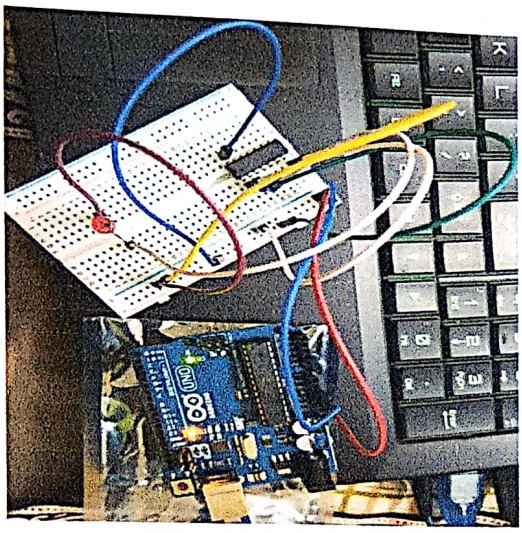


Fig.3

This figure shows the working of AND gate (CD-4081) when any of the inputs is set low. LED does not glow.

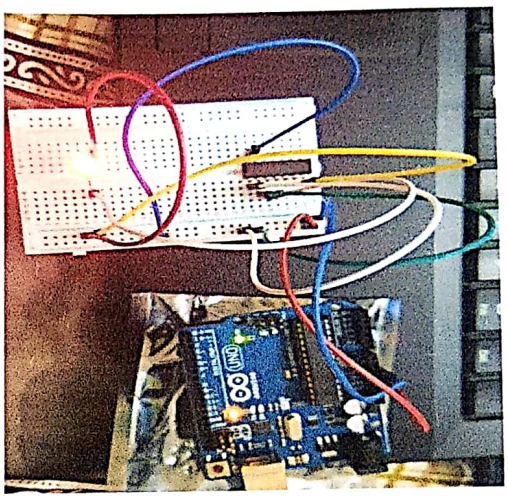


Fig.4

This figure shows the working of AND gate (CD-4081) when both the inputs are set high. → LED glows.

Reflection:

We got the verification of truth tables using AND gate and JK flip flops via hardware components.

* We learnt about the importance of clock in J/K flip flop
* We have the knowledge about set, reset conditions in J/K along with Toggling

ASSIGNMENT - 2
Study of Sensors

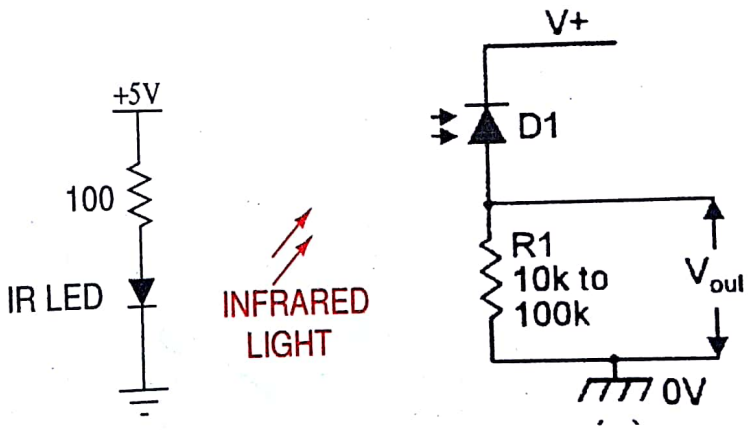
Exercise 2 – Demonstrate the working of IR sensors and receiverand displayoutput using LED.

Hardware

Bread Board, Power supply
Resistances and LED
IR transmitter and Receiver, Single core connecting wires

Theory

The figure below shows an IR pair in which IR LED emits infrared light which is received by photo diode D1 and the output voltage across resistor R1 is high. When we block the flow of light then the output voltage becomes low.



Code

Schematic:

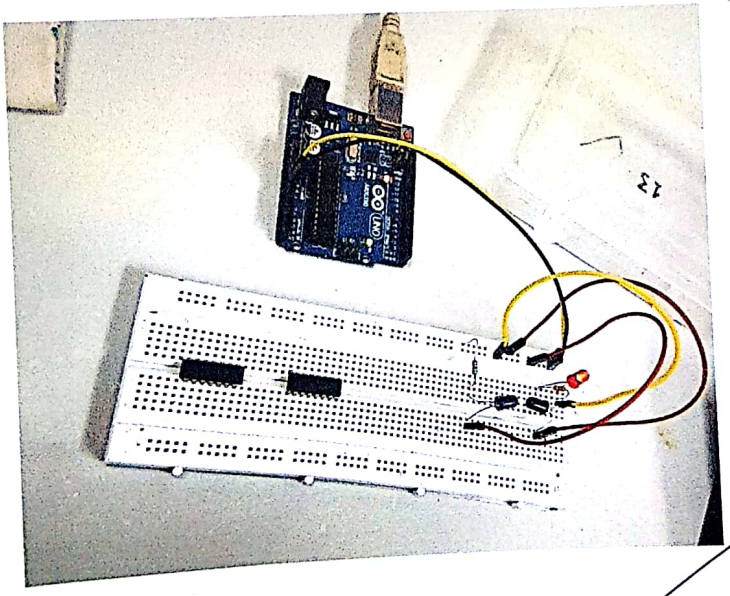


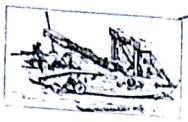
Fig.1

→ The above circuit shows the working of IR sensors. It consists of IR transmitter and receiver. Since there is no hinderence between transmitter and receiver, so the signal which is transmitted by trasmitter is received by receiver correctly is indicated by a glow in LED.

→ when hinderence, signal is not received by receiver and LED turns OFF.

Reflections

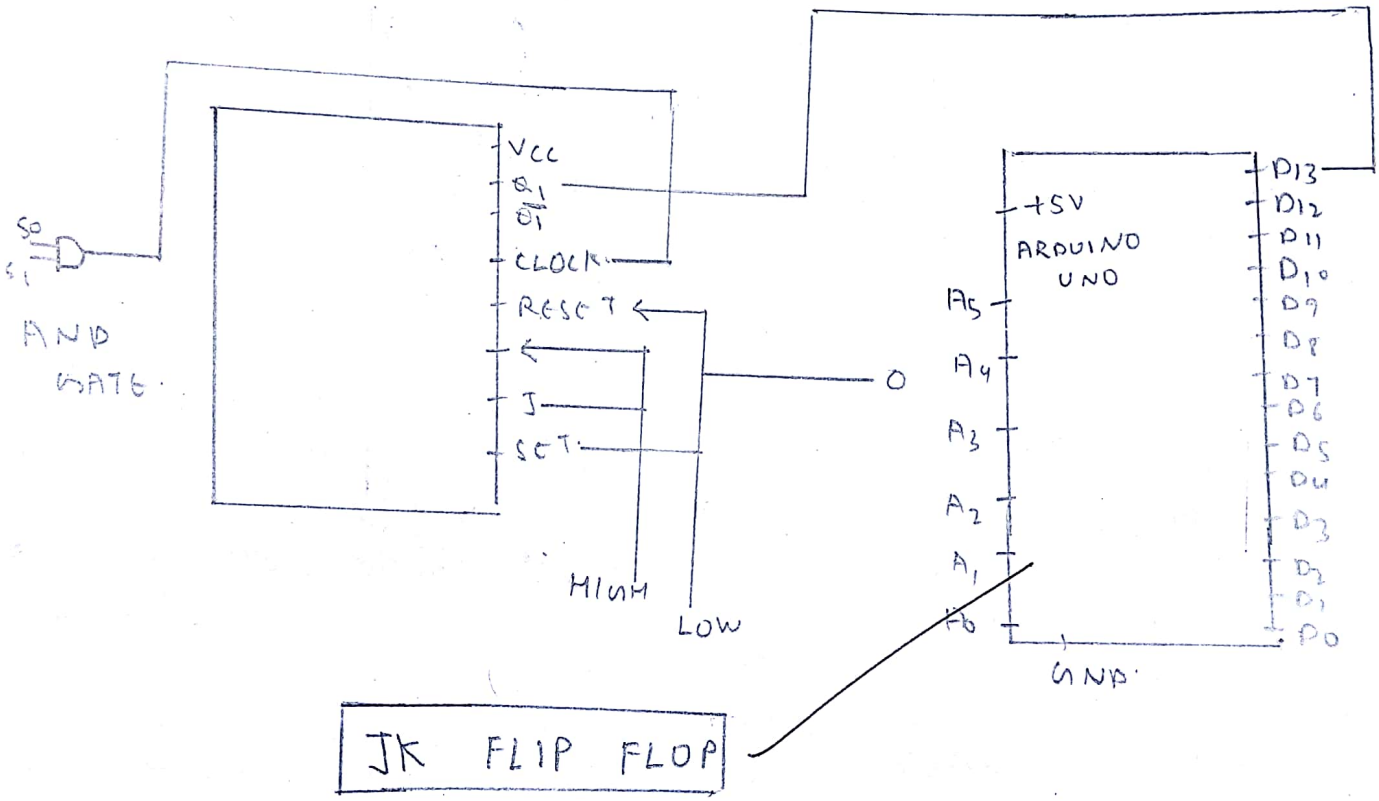
* IR sensor helped me understanding the properties of transmitter and receiver
* IR sensor works even without the software (i.e. Arduino here) coding. Just plug arduino in USB port.
* These projects will help me to do some innovations like making robots using sensors in future.



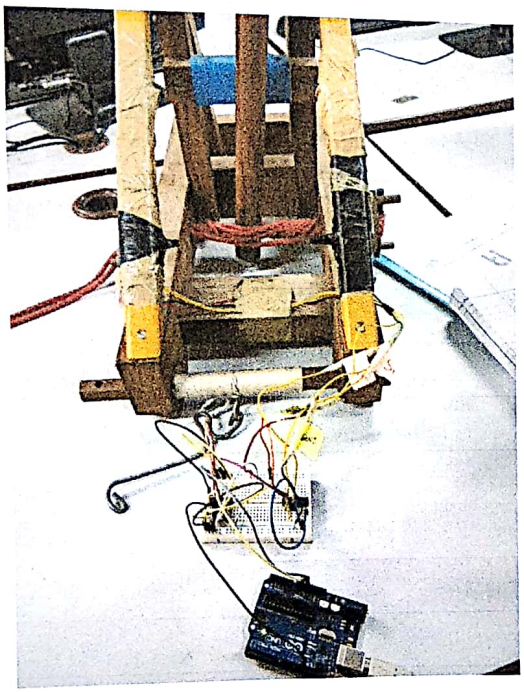
Assignment Tasks:

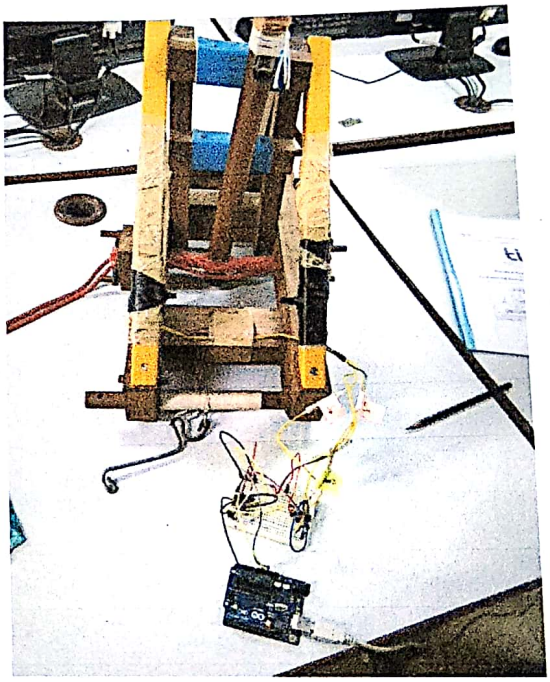
1. Use the two pair IR sensors of Mangonel to combine the two sensors output into one signal.

Circuit Diagram



Schematic





The above figure shows the IR sensors of mangonel connected with Arduino. The LED glows at the latch point. for the delay provided. Then it sets off again.

ASSIGNMENT –3(A)

Exercise 1 – To verify the functional table of CD4543

Hardware Required

- Decoder (CD4543)
- Seven Segment Display
- Single core connecting wires
- Digital Trainer Kit

Theory

The decoder (CD4543) is a combinational digital circuit that decodes an 4-bit binary input in the range 0000-1001 (BCD) in to its corresponding decimal level. Example for the binary value 0101 we need to display 5. Hence the decoder will output a HIGH on segments (a, c, d, f and g) with output a LOW on segments (b and e). The latch signal is normally connected to 5V via 10Kohm resistor as per the circuit diagram.This allows the decoder to decode the present binary input (the latch is said to be in a transparent state). When the latch is connected to 0V via the jumper provided its logic state changes to a LOW and the decoder will decode the binary input prior to the latch going low (i.e. the display is frozen when the latch is LOW).

Schematic Diagram

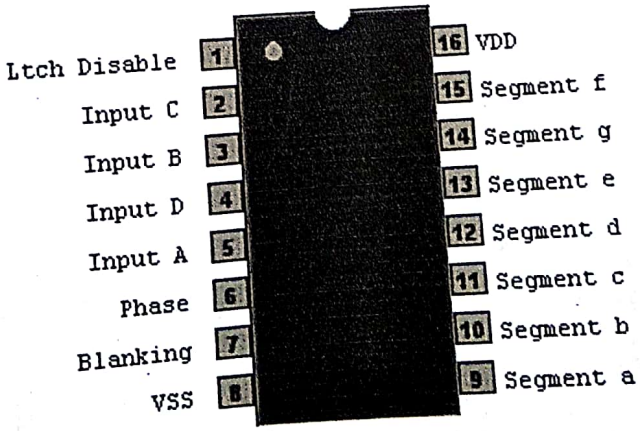


Figure 1: Pin diagram of CD4543

Truth table of CD4543B for Common Cathode Seven Segment Display

LD	BL	PH	D	C	B	A	a	b	c	d	e	f	g	DISPLAY
1	0	0	0	0	0	0	1	1	1	1	1	1	0	0
1	0	0	0	0	0	1	0	1	1	0	0	0	0	1
1	0	0	0	0	1	0	1	1	0	1	1	0	1	2
1	0	0	0	0	1	1	1	1	1	1	0	0	1	3
1	0	0	0	1	0	0	0	1	1	0	0	1	1	4
1	0	0	0	1	0	1	1	0	1	1	0	1	1	5
1	0	0	0	1	1	0	1	0	1	1	1	1	1	6
1	0	0	0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	0	0	1	1	1	1	1	0	1	1	9

Figure 2: Functional table of CD4543

ASSIGNMENT – 3(B)

Exercise 2 – BCD (binary coded decimal) to 7 Segment Display

Hardware Required

- Decoder (CD4543)
- Seven Segment Display
- Single core connecting wires
- Digital Trainer Kit
- Arduino Uno

Theory

The decoder (CD4543) is a combinational digital circuit that decodes an 4-bit binary input in the range 0000-1001 (BCD) in to its corresponding decimal level. Example for the binary value 0101 we need to display 5. Hence the decoder will output a HIGH on segments (a, c, d, f and g) with output a LOW on segments (b and e). The latch signal is normally connected to 5V via 10Kohm resistor as per the circuit diagram.

This allows the decoder to decode the present binary input (the latch is said to be in a transparent state). When the latch is connected to 0V via the jumper provided its logic state changes to a LOW and the decoder will decode the binary input prior to the latch going low (i.e. the display is frozen when the latch is LOW).

Schematic Diagram

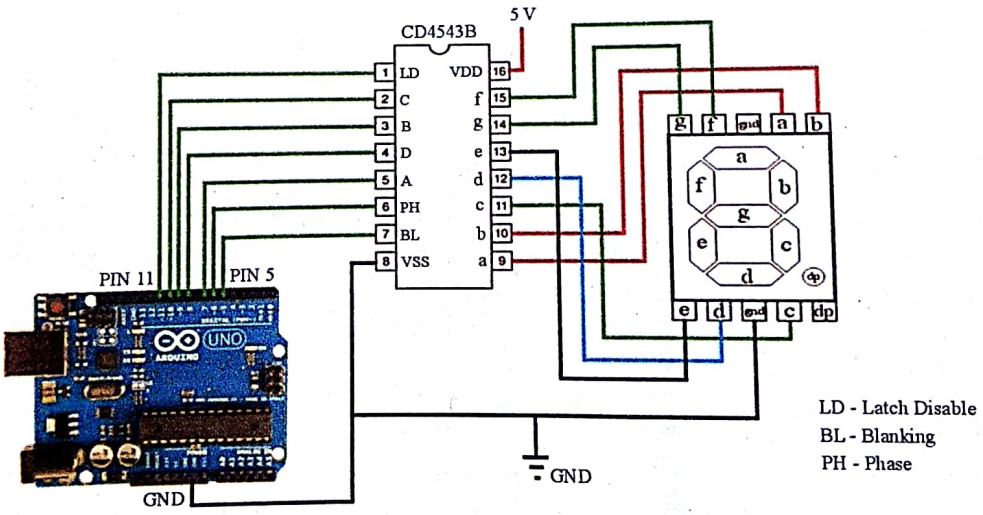
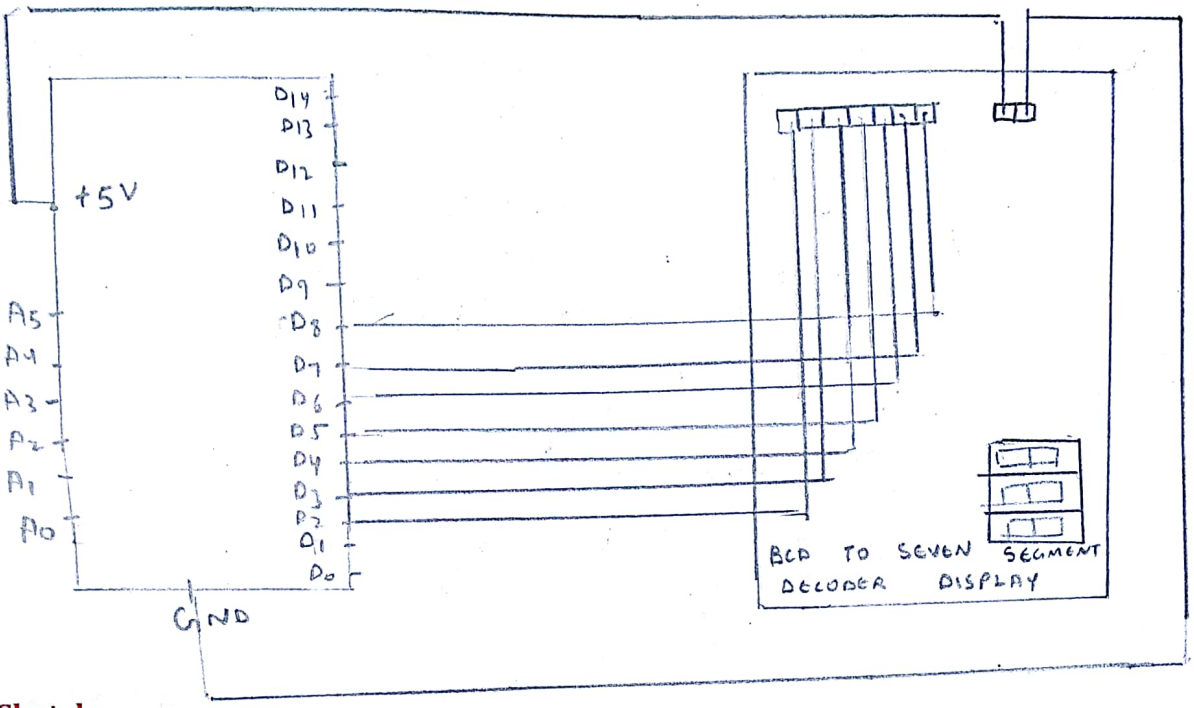


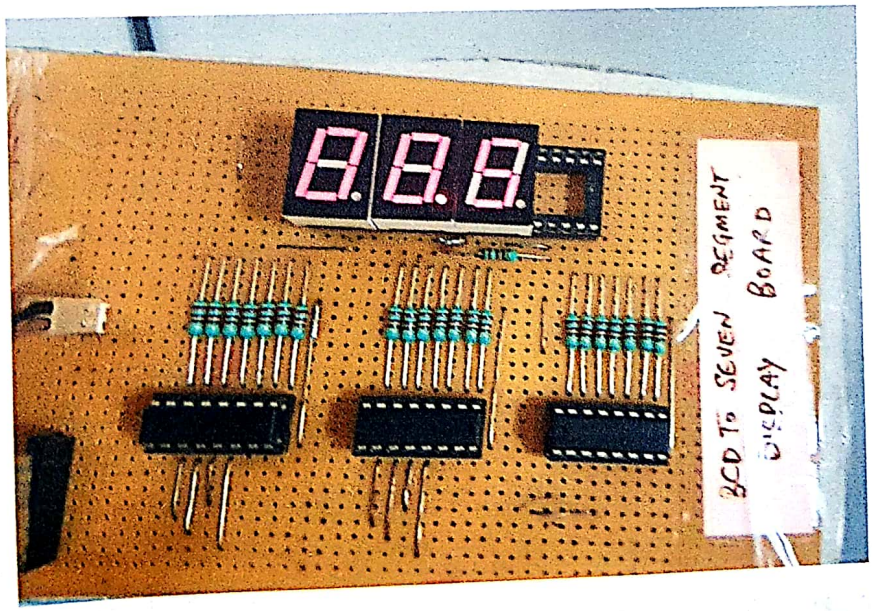
Figure 1: Connection setup for converting BCD input to seven segment output.

on segments LD

Experimental connection diagram:



Sketch:



The above figure displays the number 5 using seven segment decoder and arduino with high output on segments (a, c, d, f and g) and low output signals on segments (b and e)

Code:

```
int b, c, d, e, f, g, h;   // declaring variables
int a = 5;                 // initializing the no. to be displayed.
void setup()  // put your setup code here to run once.
{
  for (int j=2, j<=8, j++)  // running the loop.
    {pinMode (j, OUTPUT); // Assign pin no. to get output.
    }
}
  void loop ()  // put your main code here to run repeatedly.
  {
    b = a%2;
    c = a/2;
    d = c%2;
    e = c/2;
    f = e%2;
    g = e/2;
    h = g%2;
                  // operations for converting decimal to BCD.

digital Write (2, b);
digital Write (3, d);
digital Write (4, f);
digital Write (5, h);
                  // displaying BCD equivalent.
digital Write (6, HIGH);
digital Write (7, LOW);
digital Write (8, LOW);
                  // declaration of enable pin keeping 6 as high and 7, 8 as low.
}
```

* REFLECTIONS :-

⇒ The experiment teaches us about the 7 segment decoder and its operating principles.

We can input the 4 bit BCD no. and get proper output lines to form the corresponding decimal on 7 segment display using the required segments HIGH and Low.

ASSIGNMENT – 3(C)

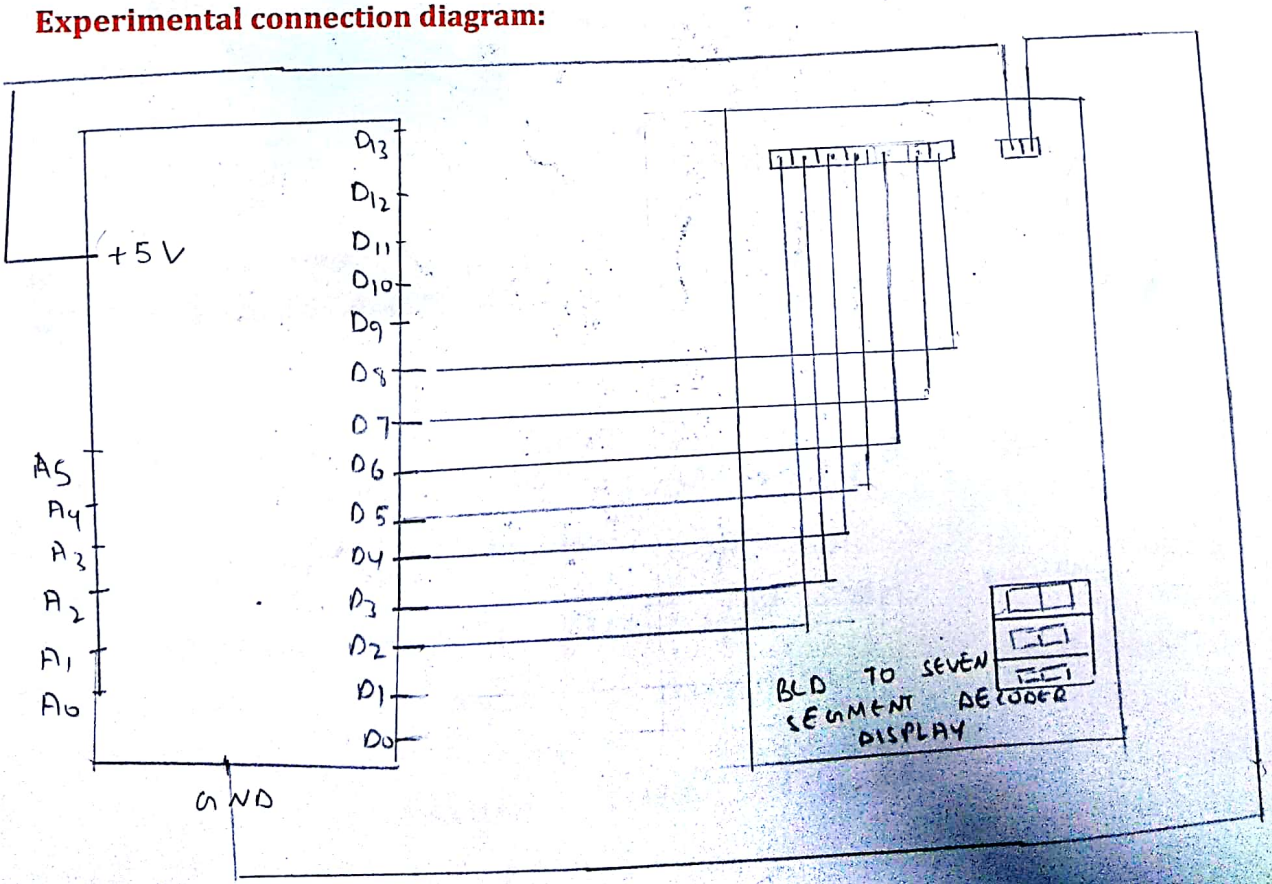
Exercise 3 –Write an Arduino sketch to make an up counter which counts from 0 to 9 & repeat it infinitely. Display the digits using BCD code on the 7-segment display on digital trainer kit.

Hardware Required

- Decoder (CD4543)
- Seven Segment Display
- Single core connecting wires
- Digital Trainer Kit
- Arduino Uno

Theory (Write the theory as per your understanding during self-effort and lab hours)

Experimental connection diagram:



Sketch:

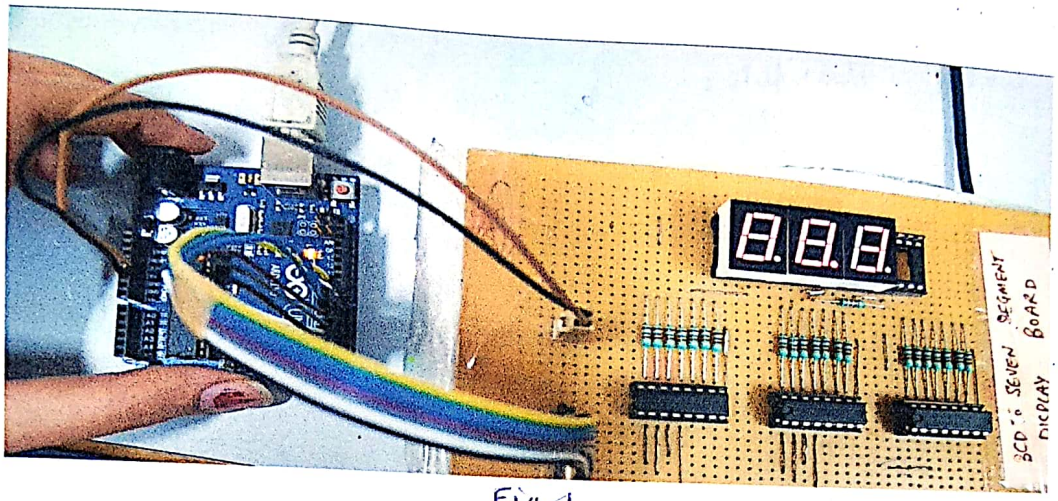


Fig-1

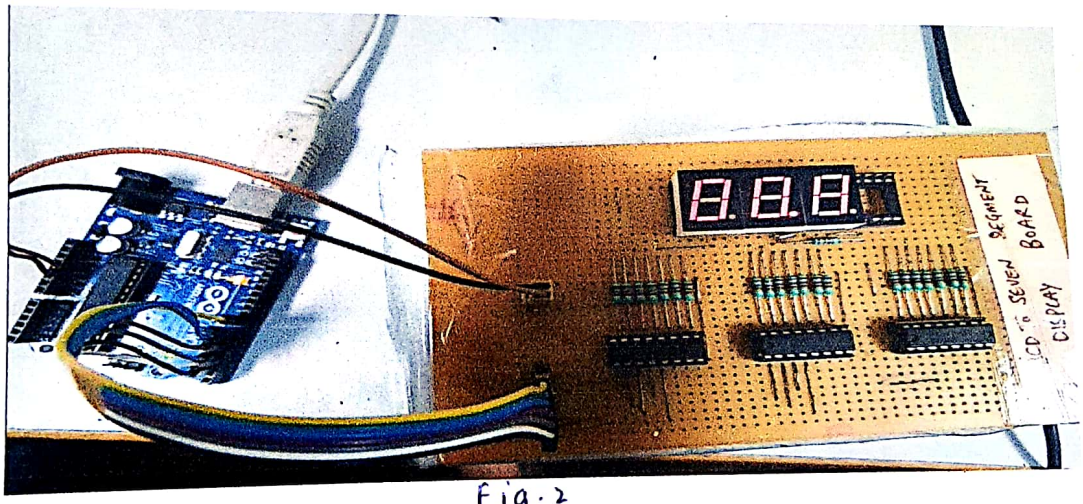
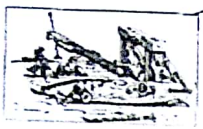


Fig-2

Fig-1 shows the no. 3 displayed on seven segment decoder.

Fig-2 shows no. 4 displayed on seven segment decoder using Arduino.

We have displayed the counter of 0-9 displayed on seven segment decoder.



Code:

```
int b, c, d, e, f, g, h; // declaring the variables
void setup()             // put your setup code below to run
{                                once.
for (int i=2, i<=8, i++)
  { pinMode (i, OUTPUT); // Assign pin no. to get output
  }
}
void loop() // put your main code to run repeatedly.
{ for (int a=0, a<=9, a++) // executing for loop initializing
                           from 0 to 9.
  { b=a%2;
    c=a/2;
    d=c%2;
    e=c/2;
    f=e%2;
    g=e/2;
    h=g%2;
digitalWrite(2,b);
digitalWrite(3,d);
digitalWrite(4,f);
digitalWrite(5,h);
digitalWrite(6,HIGH); digitalWrite(7,LOW); digitalWrite(8,LOW);
                      // sets 6th as High and 7,8 as Low.
  delay(2000);
                // wait for 2 seconds
}}
```

// operation for converting decimal to BCD

// displaying BCD equivalents

Reflections:

⇒ Importance of seven segment decoders which finds application in calculator, digital clocks.

⇒ we understood the conversion from decimal into BCD

⇒ we understood how individual segments can be set high and low.

⇒ Execution of for loop is implemented.

Assignment Tasks:

- Given a time in microsecond which is a 5-digit integer. Displayed the equivalent time in millisecond using Arduino on two digit seven segment display.

Schematic Diagram

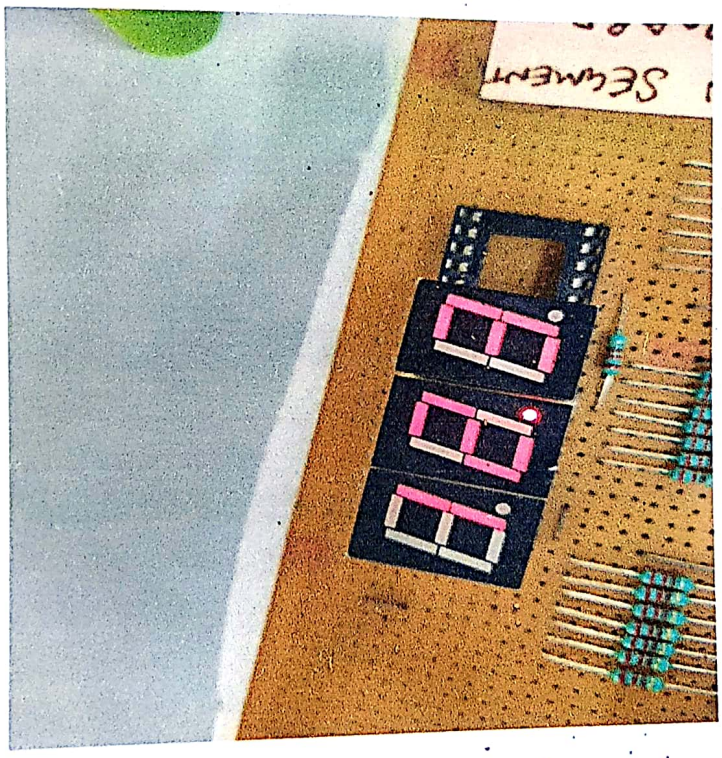
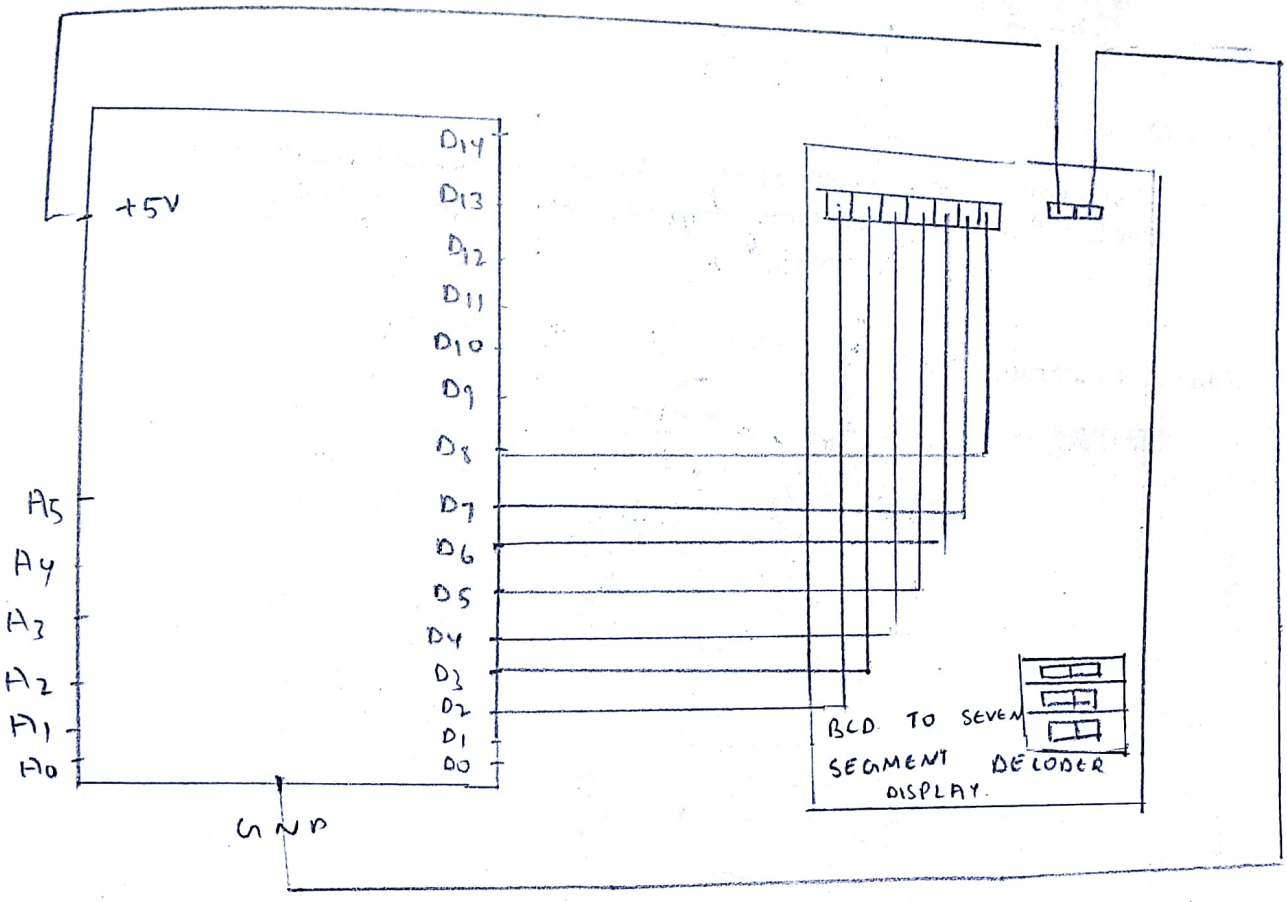


Fig.1

Fig. 1 displays the no. 123 using seven segment decoder. out of the inputted 5 digit no. (12345) using Arduino.

EXPERIMENTAL CONNECTION DIAGRAM:



Code:

```
double x = 12345; // initializing the 5 digit number.
void setup() // put your setup code here to run once:
{
for (int z=2, z<=8, z++)
  {
  pinMode (z, OUTPUT); // Assign pin no. to get output.
  }
}
void loop() { // put your main codere to run repeatedly.
{
int a, b, c, d, e, f, g, h, R, P, q, r, s, t, u;
                                    // declaring variables.
t = x/100;
f = t %. 10;
g = f %. 2;
f = f/2;
h = f %. 2;
f = f/2
k = f %. 2;
f = f/2;
```

// operations for converting decimal to BCD at the 100^th's place.

Reflections:

```
digitalWrite (8, HIGH);
digitalWrite (12, g);
digitalWrite (11, h);
digitalWrite (10, k);
digitalWrite (9, f);
```

// setting enable 8 as HIGH.
// displaying BCD equivalent.

```
digitalWrite (8, LOW); // setting 8th pin as Low.
delay (1000); // pin is set low for 1 second.
```

```
d = d/10;
e = 4%10;
a = e%2;
b = e%2;
e = e/2;
c = e%2;
e = e/2;
```
// operations for 10's place display of seven segmented display.

```
digitalWrite(7, HIGH); // sets 7th pin as HIGH.
digitalWrite(12, a);
digitalWrite(11, b);
digitalWrite(10, c);
digitalWrite(9, e);
digitalWrite(7, LOW); // sets 7th pin as LOW.
delay(1000); // waits for 1 second.
```
// displaying BCD equivalents.

```
u = u/10;
p = u%10;
q = p%2;
p = p/2;
r = p%2;
p = p/2;
s = p%2;
p = p/2;
```
// operations for one's place display of seven segmented display.

```
digitalWrite(8, HIGH); // sets 8th pin as HIGH.
digitalWrite(12, q);
digitalWrite(11, r);
digitalWrite(10, s);
digitalWrite(9, p);
digitalWrite(8, LOW); // sets 8th pin as Low
delay(500); // waits for 0.5 seconds.
}
```
displaying BCD equivalents.

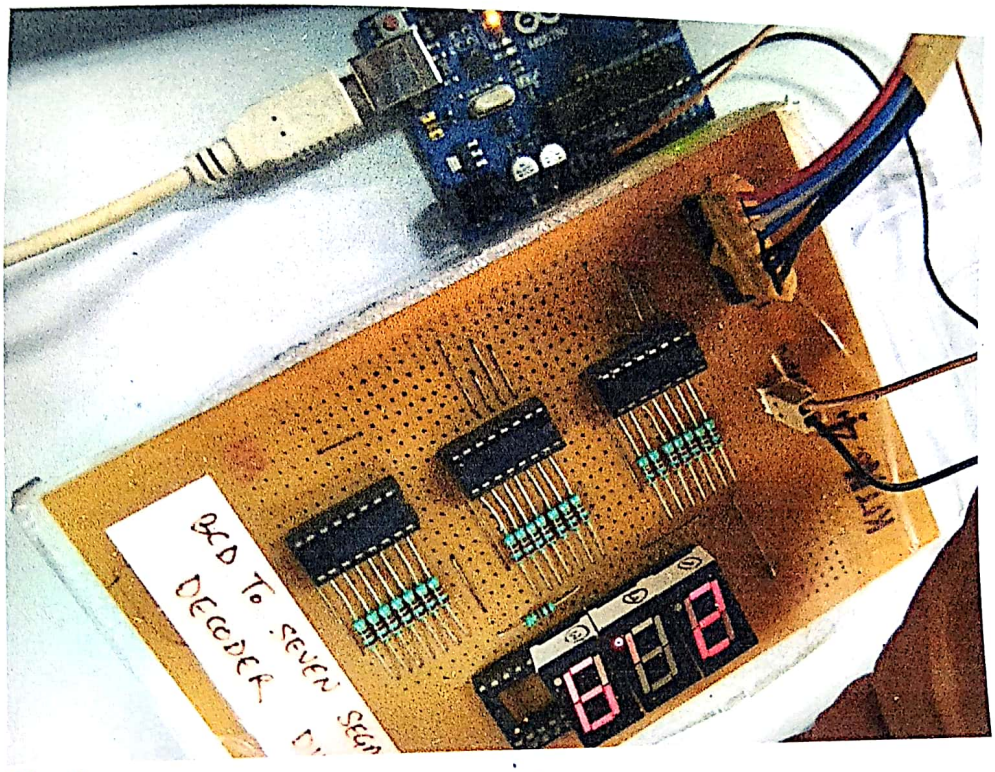
REFLECTIONS

⇒ we learn the use of 2 digit seven segment display using arduino.

⇒ Application of multiple loops to get the required no. is implemented.

- Redesign the Exercise 3 from 00 to 99.

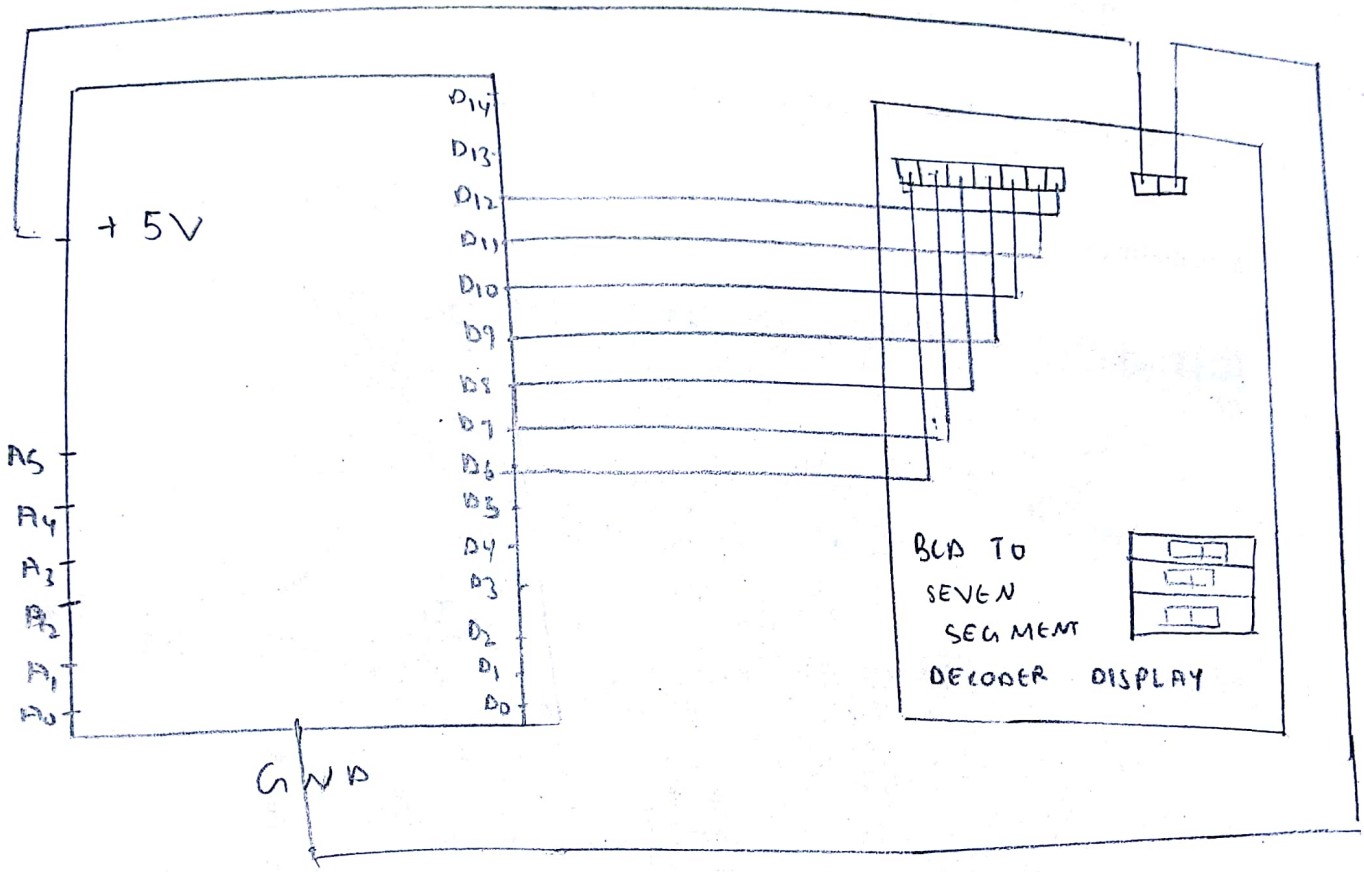
Schematic Diagram



Fig, 1

The above figure displays all the no's from 00 to 99 using seven segment display with the help of Arduino.

EXPERIMENTAL CONNECTION DIARGAM:



Code:

```
void setup() // put your setup code here to run once.
{
  pinMode (12, OUTPUT);
  pinMode (11, OUTPUT);
  pinMode (10, OUTPUT);
  pinMode (9, OUTPUT);
  pinMode (8, OUTPUT);
  pinMode (7, OUTPUT);
  pinMode (6, OUTPUT);
}
```

Assign pin no. to get output

```
void loop() { // put your main code to run repeatedly.
int i, j;
for (j=0, j<=9; j++) // applying loop.
{
  int a, b, c, d, e, f, g, h, k; // declaring variables.
  f = j;
```

Reflections:

```
  g = f%2;
  f = f/2;
  h = f%2;
  f = f/2;
  k = f%2;
  f = f/2;
```

operations for converting decimal to BCD.

```
digitalWrite (6, HIGH); // sets 6th pin as HIGH.
digitalWrite (12, g);
digitalWrite (11, h);
digitalWrite (10, k);
digitalWrite (9, f);
digitalWrite (6, LOW); // it sets 6th pin as LOW
delay (500); // waits for 0.5 second.
```

displays BCD equivalent.

```
for (i = 0, i <= 9, i++)  // nesting of loop.
  {
    e = i;
  a = e % 2;
  e = e/2;
  b = e % 2;
  e = e/2;
  c = e % 2;
  e = e/2;
digital Write (7, HIGH);  // sets 7th pin as high.
   digital Write (12, a);
    digital Write (11, b);
     digital Write (10, c);
      digital Write (9, e);
       digital Write (7, LOW);  // sets 7th pin as Low.
  delay (500);  // waits for 0.5 second.
}
}}
```

// operations of converting decimal to BCD using seven segment decoder.

// displays BCD equivalents.

* REFLECTIONS *

* we get knowledge about 2 digit seven segment display.
* Enhances the arduino coding skills linked with the hardware.
* Enlightens the LED properties of different segments in seven segment decoder.