Embedded Systems Interfacing

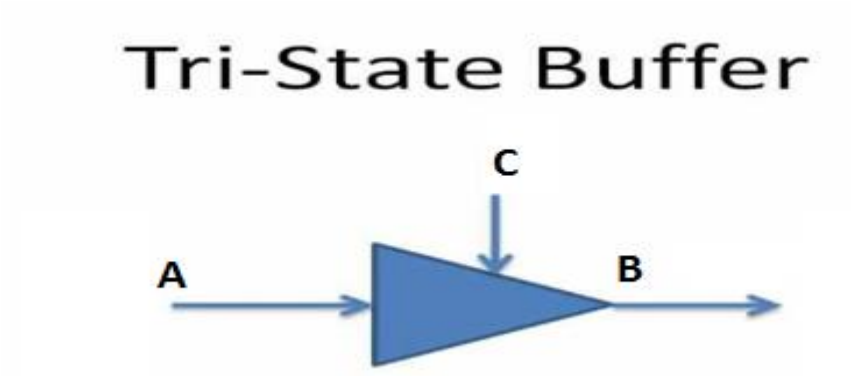
Lecture one

Digital Input Output Part 1

Digital Input Output

A Digital Input Output is a peripheral that deals with digital signals, either by generating a digital signal (***Output Mode***) or by receiving it (***Input Mode***).

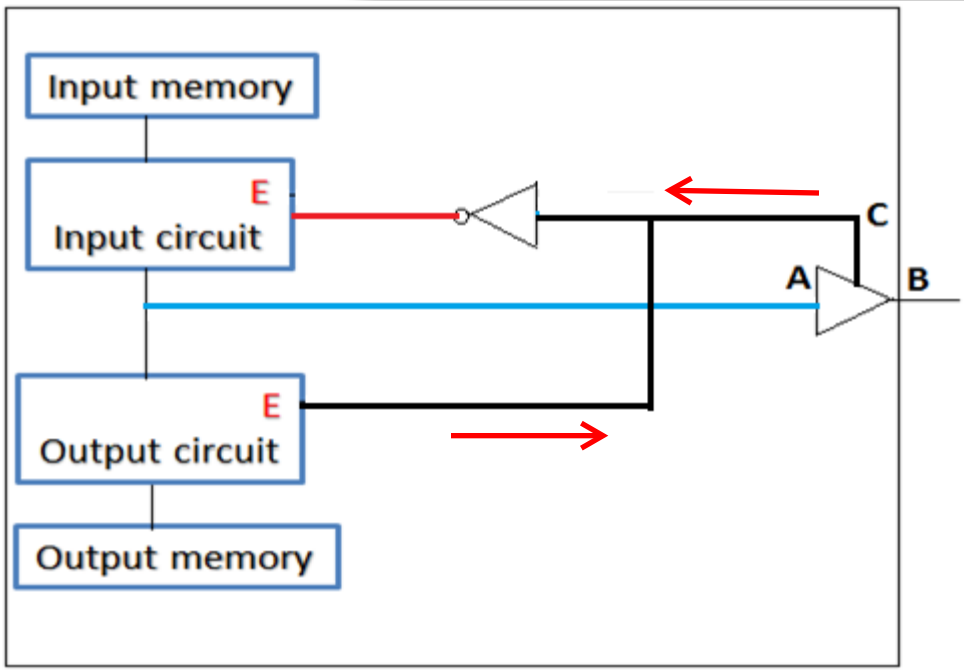
The basic block unit for the DIO pin is the ***Tri-State Buffer***. Any DIO pin is consisting of a Tri-State buffer as a main component. The Tri-State buffer controls the direction of the data, A to B or B to A.



C	Output
1	**A → B**
0	**B → A**

DIO Block Diagram

Writing 0 To C, Enables the Input Circuit and make the Buffer direction to (A <- B) which makes the PIN in ***Input Mode***



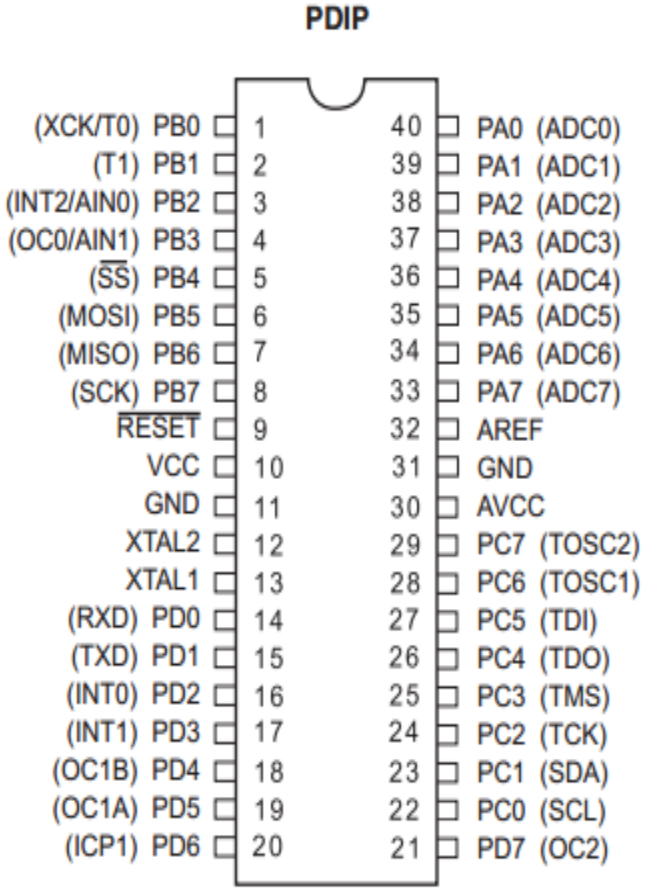
Writing 1 To C, Enables the Output Circuit and make the Buffer direction to (B <- A) which makes the PIN in ***Output Mode***

AVR Microcontroller

In our course we will use Microcontroller AVR Atmega32. It has 32 DIO pins grouped as following:

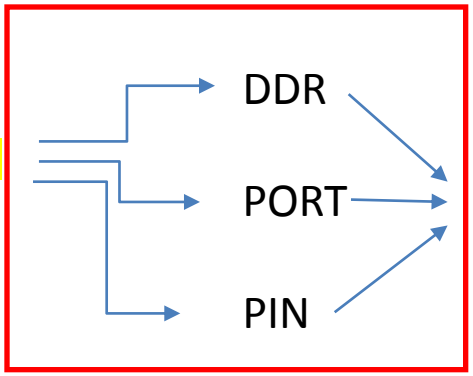
1- **PORTA** has 8 DIO Pins from ***A0*** to ***A7***
2- **PORTB** has 8 DIO Pins from ***B0*** to ***B8***
3- **PORTC** has 8 DIO Pins from ***C0*** to ***C8***
4- **PORTD** has 8 DIO Pins from ***D0*** to ***D8***

Each pin can work either in input mode or output mode.



AVR DIO

Every port has 3 control registers



The size of each register is 8 bit, every bit ***corresponding*** to 1 Pin of the port

1- DDR (Data Direction Register) in this register we can define the pin is output or input

- Set **0** → Input
- Set **1** → Output

2- PORT : This register is used in output mode to set the digital output value

- set **1** → this pin carry 5v
- set **0** → this pin carry 0v

3- PIN: we use this register in case the pin is defined input

- if **1** → The Pin is connected to 5v
- if **0** → The Pin in connected to 0v

AVR DIO Example

Example 1

Configuring Pin A0 and output 5V

```
DDRA=0b00000001 ;
PORTA=0b00000001 ;
```

Example 2

Configuring All PORTD Output, lower half connected to 5V and upper half connected to ground

```
DDRD=255;
PORTD=0x0f;
```

Remember Numbering Systems in C

Remember the major three numbering systems used in C:

- Binary : PORTA=0b11111111;
- Decimal : PORTA=255;
- Hexadecimal : PORTA=0xff ;

All of these Statements are Equivalent

Interfacing LEDs

LED Definition

Light Emitting Diode is an electrical element that emits light by supplying a voltage difference between its terminals

LED Connection:

The LED has two pins, positive and negative one.
In your kit there are 8 LEDs all of them are common ground..

Coding led :

DDRA=0b00000001 ; → Configure Pin A0 in Output mode

PORTA=0b00000001 ; → Set A0 to digital 1 i.e 5V

Now If we are connecting the negative pin of led to ground and connecting positive pin of led to A0 , the led will be lighted up

then

PORTA=0b00000000 ; → Set A0 to digital 0 i.e 0V

Now the led will be closed up

The Super Loop

Any C project in Embedded Systems application shall have an infinite loop called the ***super loop***. This loop is a ***must*** even if you we will leave it empty !

This loop prevents the program counter ***(PC)*** from continues incrementing over the flash memory and execute a garbage code. i.e. the while(1) represents the end of the code.

```
void main(void)
{
    /* Initialization  Part */


    /* The Super Loop */
    while (1)
    {

    }
}
```

IMT SCHOOL

LAB 1

Write a C code to turn on LED on Pin A0

Time To Code

Using Delay Function

Busy Loop Delay

Software Technique the use a loop with effect just to halt the processor for certain time. We will use a library called *"avr/delay.h"* that provides two basic functions:

1- ***_delay_ms*** (*_value_in_ms*) /* Apply a delay in milli seconds */
2- ***_delay_us*** (*_value_in_us*) /* Apply a delay in micro seconds */

Note

Before using the delay library, we have to define our system frequency by writing this command:

```
#define F_CPU   12000000  /* Define a CPU frequency of 12 Mega Hertz */
```

LAB 2

Write a C code to turn on LED on Pin A0 for 1 second and then turn it off.

Time To Code

LED Blinking

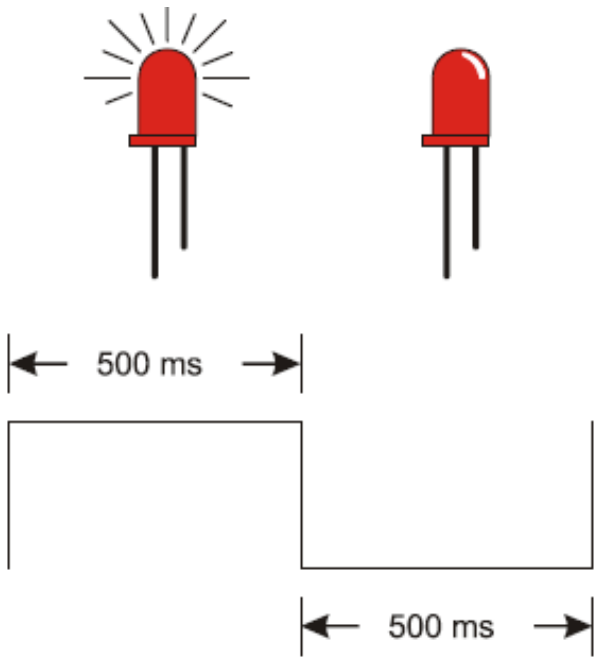
LED Blinking Algorithm

```
/* Loop forever */
while (1)
{
   /* Turn LED on */
   PORTA = 0x01;

   /* Apply 0.5 Second Delay */
   _delay_ms(500);

  /* Turn LED off */
   PORTA = 0x00;

   /* Apply 0.5 Second Delay */
   _delay_ms(500);
}
```



LAB 3

Write a C code to blink a LED Every 1 second

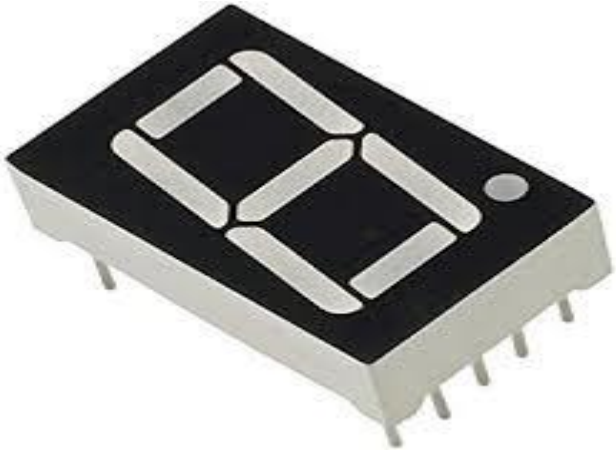
Time To Code

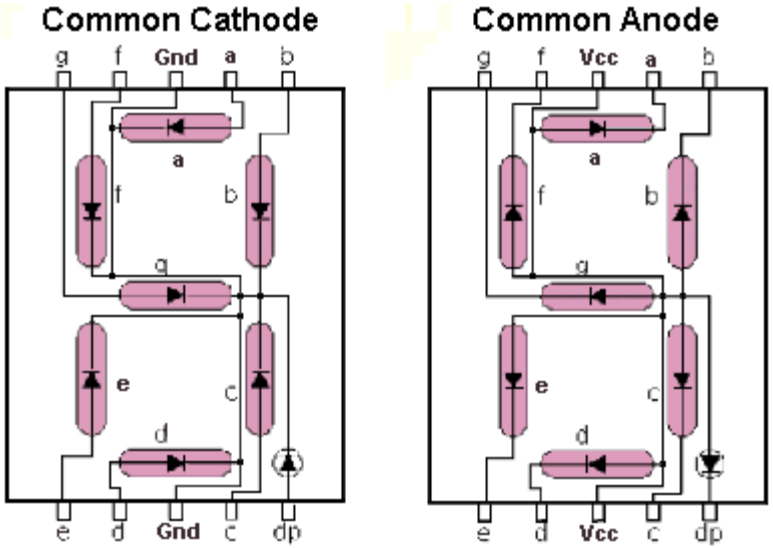
LAB 4

Write a C Code that apply Some LED animations

Time To Code

Interfacing 7-Segments



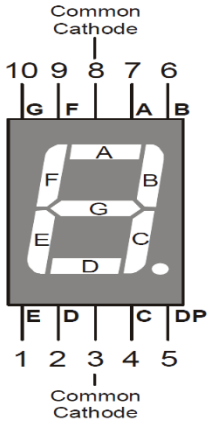


Interfacing 7-Segments

Coding of 7-Segments :

7–segment is common cathode :

Assuming Connecting the 7-Segment lines (a to g) to PD0 to PD6 in the microcontroller kit



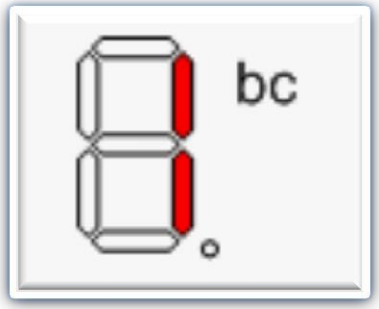
DDRD=255 ;

Configure PORTD in output mode

PORTD=0b00000110 ;

Set second and third pin in PORTD to carry 5v.

output on 7-segement



then

PORTD=0b01011011 ;

Set D0 , D1 , D3 , D4 & D6 in PORTD to carry 5v that connected by A,B,D,E,G

output on 7-segement

7-Segment Truth Table

s	BCD	G	F	E	D	C	B	A
0	0000	0	1	1	1	1	1	1
1	0001	0	0	0	0	1	1	0
2	0010	1	0	1	1	0	1	1
3	0011	1	0	0	1	1	1	1
4	0100	1	1	0	0	1	1	0
5	0101	1	1	0	1	1	0	1
6	0110	1	1	1	1	1	0	1
7	0111	0	0	0	0	1	1	1
8	1000	1	1	1	1	1	1	1
9	1001	1	1	0	1	1	1	1

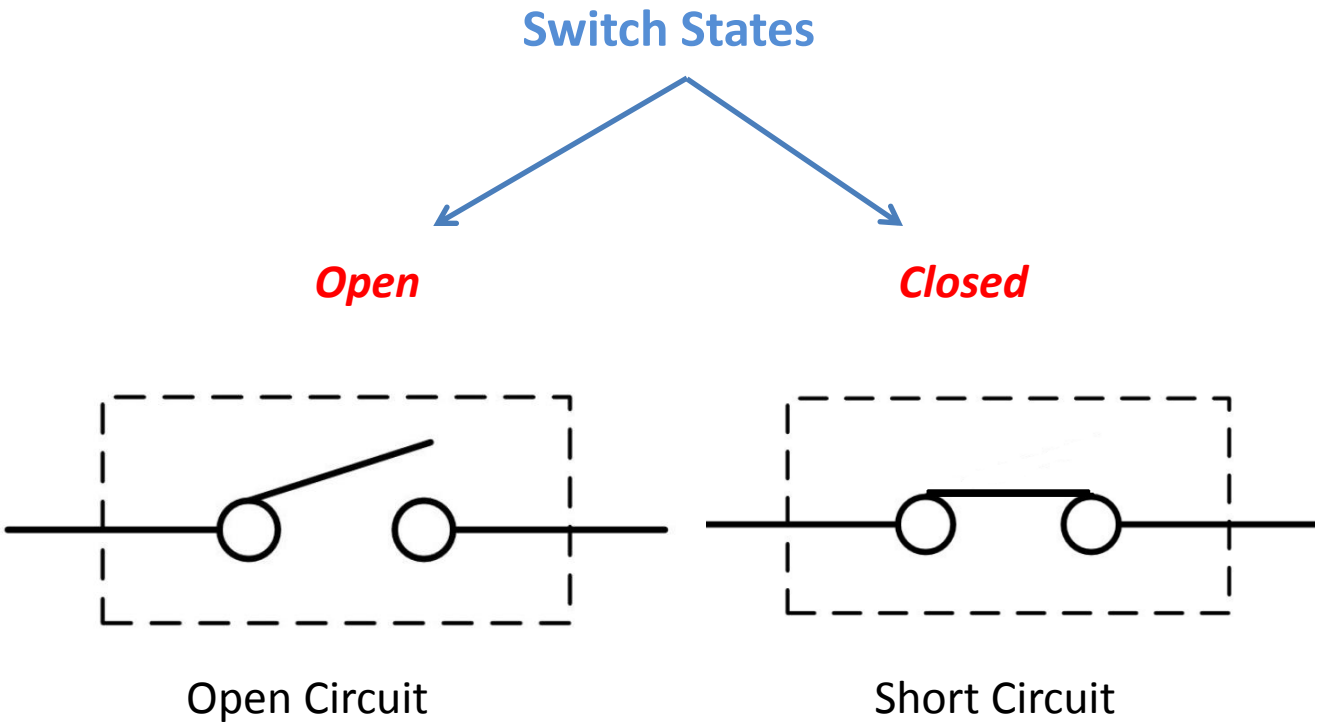
LAB 5

write a code to display on 7-segement numbers from 0 to 9 with delay 1 second before changing number.

Time To Code

Mechanical Switch

Mechanical switch
is an electrical component that can connect or break an electrical circuit.



Tactile switch

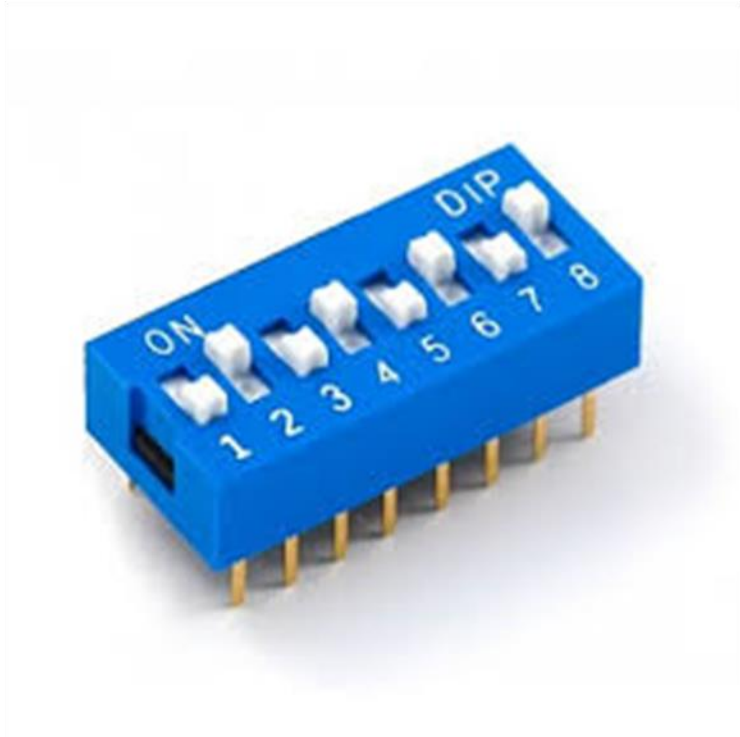
Push Button

Paddle Switch

Rocker Switch

Toggle Switch

DIP Switch



Thumbwheel Switch

Limit Switch

Slide Switch

Rotary Switch

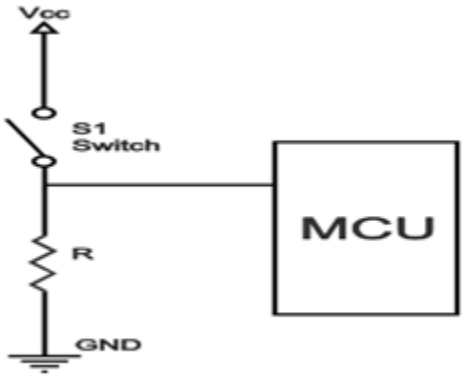
Reed Switch

Knife Switch

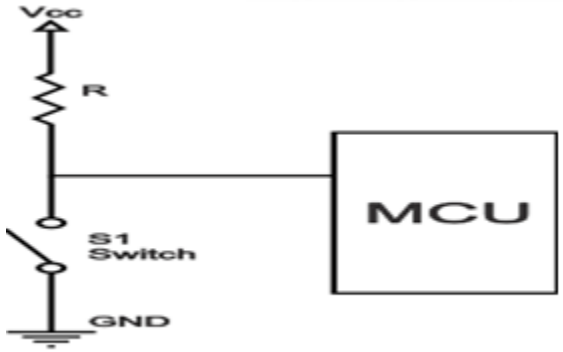
Key Switch

Interfacing Mechanical Switch

Switch shall be connected by pull up or pull down resistor to avoid short circuits.



Pull-down resistor

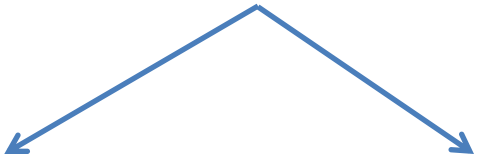


Pull-up resistor circuit

Interfacing Mechanical Switch

In AVR Microcontroller, all DIO pins have internal pull up resistors that can be activated or not.

Input Digital Pin



Floating

/* Configure PIN as input */ DDR -> 0

In this state, the DIO pin has 3 states, 0 when connected to GND, 1 when connected to VCC, floating when not connected to anything which may be read as 0 or as 1 !

Note Never let an input pin as floating to avoid noise affection.

Internal Pull Up

DDR -> 0 /* Configure PIN as input */

PORT -> 1 /* Activate Internal Pull up */

In this state, the DIO pin has 2 states only, 0 when connected to GND, 1 when connected to VCC or when not connected to anything.

Reading Input PIN

The registers ***PINA***, ***PINB***, ***PINC*** and ***PIND*** are used to check the status of the input pins. If the corresponding bit for a certain pin is ***0***, then the pin is connected to ***GND***. If the corresponding bit for a certain pin is ***1***, then the pin is connected to ***VCC***.

```
/* Check if Pin A0 is conncted to GND  */
if ( (PINA & 0b00000001) == 0)

/* Check if Pin B3 is connected to VCC */
if ( (PINB & 0b00001000) != 0)
```

LAB6

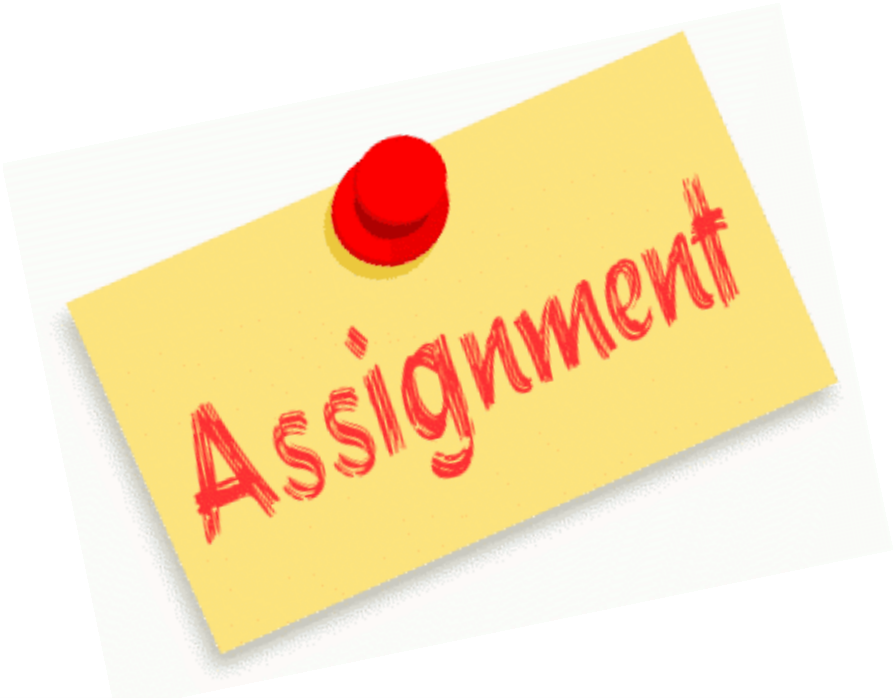
Write a code that uses a DIP switch to control a string of 8 LEDs. When the DIP switch is On the LED string shall be flashing every 500 ms. When the DIP switch off the LED string shall be also off.

Time To Code

The End …

Assignment 1

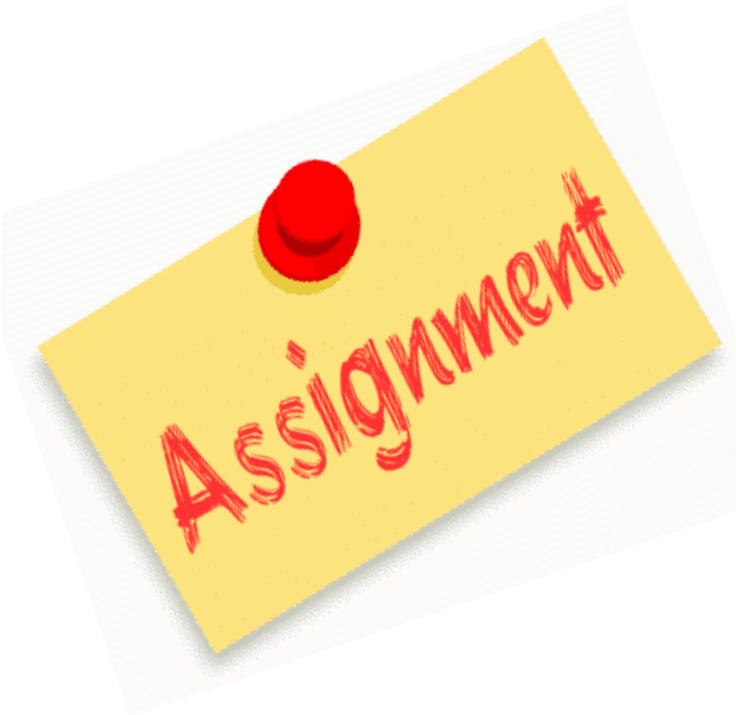
Write a C code that simulate the traffic lightening system:
1- Turn On Green LED for 10 seconds
2- Turn On Yellow LED for 3 seconds
3- Turn On Red LED for 10 seconds
4- Apply these forever while counting the seconds down on a 2 7-segment displays.



Assignment

Write a C code that apply 8 different animations on 8 LED string based on the value of 3 way DIP Switch as following:

DIP value	LED Action
1	Flashing every 500 ms
2	Shifting Left every 250 ms
3	Shifting Right every 250 ms
4	2-LEDs Converging every 300 ms
5	2-LEDs Diverging every 300 ms
6	Ping Pong effect every 250 ms
7	Incrementing (Snake effect) every 300 ms
8	2-LEDs Converging/Diverging every 300 ms



IMT SCHOOL

www.imtschool.com

ww.facebook.com/imaketechnologyschool/