Experiment - 8   **Interrupt and Serial display**   **Monsoon 2018**

An interrupt is a process of forcing a microcontroller to temporarily suspend the current job and perform a higher priority task. Once the new request is served, the microcontroller resumes the original task. Interrupts are powerful tools in embedded system design, and special cares must be taken while implementing them. In this experiment you will learn how to use external interrupt of Aruidno

Arduino has several types of interrupts. An interrupt is a process by which arduino stops its regular task or stop its looping and go to interrupt function to complete its given interrupt function task. External interrupt created externally. There is only two external interrupt pin in Arduino. They are Digital Pin 2 and Digital Pin 3.

After initialization of external interrupt if there is any change in signal in this pin. Then that will create external interrupt.

"attachInterrupt()" is the function used to create external interrupts.

attachInterrupt(Pin,Func,Mode).

Pin : pin number to which interrupt is attached

Func : the function which will be called when interrupt is triggered

Mode : triggering mode for interrupt this can be

> (1)LOW: to trigger the interrupt whenever the pin is low,
>
> (2) CHANGE: to trigger the interrupt whenever the pin changes the value
>
> (3) RISING: to trigger when the pin goes from low to high,
>
> (4) FALLING: for when the pin goes from high to low.

This will call the function " Func " if the Pin will satisfy the condition in Mode

You have to make the value of counter go up every time you press an external push button. The main problem you will face is bouncing of push button switch. So when you press the push button switch once counter value will increase by one or more than one. Read about analog de-bouncing circuit and come prepared with a circuit.

Pseudo Code

```
//initialize global variables

void setup{

        //initialize Serial Monitor and AttachInterrupt()


Void loop{

        //print counter value

}
Void func{


        //code for what interrupt should do
}
```

Debouncing circuit: