NORMAL MODE :-

In normal mode the code first being flashed in cpu of AVR and then executed sequentialy.

To get the pulse of desired width(some duty) there are two methods :

1.Using _delay_ms() function :

First make the pin high or low and then use appropriate delay (coresponding to that of duty) .

This method is not accurate and inefficient because delay function itself reqiure some clock tics and and its measurement are not accurate always what we want .

Also , delay function make the CPU busy for calculating loops and thus wasting the time or using it inefficiently .

2.Using timers and interrupt combination :

AVR(Atmega328p) has three timers (timer0 , timer2 --> 8bit and timer1-->16bit) which produces interrupts on campare and match output or at overflow of register .

I am using timer0 for this setup experiment which is 8 bit , thus it can count from 0 to 255 .

Now , clock frequency is 16 Mz and I am using 1024 prescaler to reduce its frequency . So, overall I have 16Mz/1024 = 15,625 Hz (because I don’t need much precesion for just blinking led on-off for secounds) .

Calculatiog number of ticks require for making led high for 4 sec and making it low for 1 sec .

Time for 1 tick = 1/15625 sec ;
So, number of tics required for 4 sec is 4*15625 ;

But our regsister can count upto 8bit (256 tics) .

so , defining a term overflow , so one overflow = 256 tics .

So , number of overflow = (4*15625)/256 = ~ 244 .... 0.1406 error which is (1/15625)*256*0.1406 = 0.0023036 sec error

Similerly , for led to be off for 1 sec , number of overflow required = ~61

ISR (interrupt service routin) :

Whenever there is overflow a TIMER0_OVF_vect is fired and reading that interupt in my ISR function , i am can number of overflows .

Unfortunately , i am not able to use over_flow interupt , so i used comapre and match interrupt . Which will compare time timer tick and once it reaches the 255 it wire fire an interupt vector .

To run the code in AVR i am using avr-gcc and for debegur i am usinhg gdg-avr and for flashing the code into avr i am using avrdude .

The commands for them are :

```
avr-gcc -Os -DF_CPU=16000000UL -mmcu=atmega328p -c -o intr.o
intr.c

avr-gcc -mmcu=atmega328p intr.o -o intr

avr-objcopy -O ihex -R .eeprom led intr.hex

avrdude -F -V -c arduino -p ATMEGA328P -P /dev/ttyACM0 -b 115200
-U flash:w:intr.hex
```

The port ttyACM0 , can be ttyACM1 or anything else and this can be checked in the foder containg usb port list in linux it is ...... ls /dev/

Pulse Generator Setting			Description of led behavior
Pulse	Duty cycle (%)	Sample Time	
2 sec	50	Nil	Precise measurement with error in millisecs (~ 2 milli sec from cal.)
2 sec	25	Nil	||
4 sec	80	Nil	||

Sample time :
The sample time of a block is a parameter that indicates when, during simulation, the block produces outputs.

Its simulation related term , physicaly if we make it too high then our output will be dalayed because we are not allowing to produce the output .

And if its two low the block will try to generate output but the the output is not ready .