Lab report 04

**Interfacing Push buttons to 8051 using Interrupts**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**307L-MBSD LAB**

**Department of Computer System Engineering**

**University of Engineering and Technology Peshawar**

Submitted to: **Engr. Shahzada Fahim Jan**

Submitted by: **Muhammad Saad**

Reg ID: **21PWCSE1997**

**DCSE, Batch 23, Section “B”**

Date: **Sat, 16th March, 2024**

**Lab 04**

**Interfacing Push buttons to 8051 using Interrupts**

**Objective:**

* To Learn how to use interrupt in 8051 micro-controller

**Components needed for this lab:**

* Keil µVision IDE
* Proteus Software
* 8051 Micro-controller

**8051 Micro-controller:**

* 8051 have 40 pins dedicated for various functions such as I/O, -RD, -WR, address, data, and interrupts.
* The 8051 has an on-chip oscillator but requires an external clock to run it. A quartz crystal oscillator is connected to inputs XTAL1 (pin19) and XTAL2 (pin18). The quartz crystal oscillator also needs two capacitors of 30 pF value



Fig 01: 8051 Pin Diagram

**Questions:**

**1: Use Polling and Interrupt side by side.**

#include <reg51.h>

#include <stdio.h>

sbit P\_button = P2^0;

sbit P\_LED = P2^1;

sbit I\_button = P3^2;

sbit I\_LED = P3^4;

void e\_i\_() interrupt 0

{

int i=0;

for(i=0; i<1000; i++)

I\_LED ^=1;

}

void main()

{

I\_button = 1;

P\_button = 1;

EA = 1;

EX0 = 1;

while(1)

{

if(P\_button==0)

P\_LED = 1;

else if(P\_button==1)

P\_LED = 0;

}

}

8051 C Code to show difference between polling and interrupt

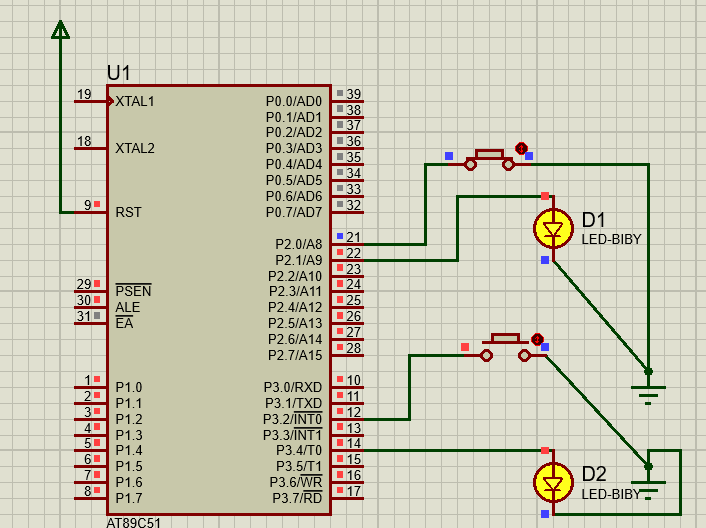


Fig 02: LED on port 2 use polling while on port 3 it use interrupt

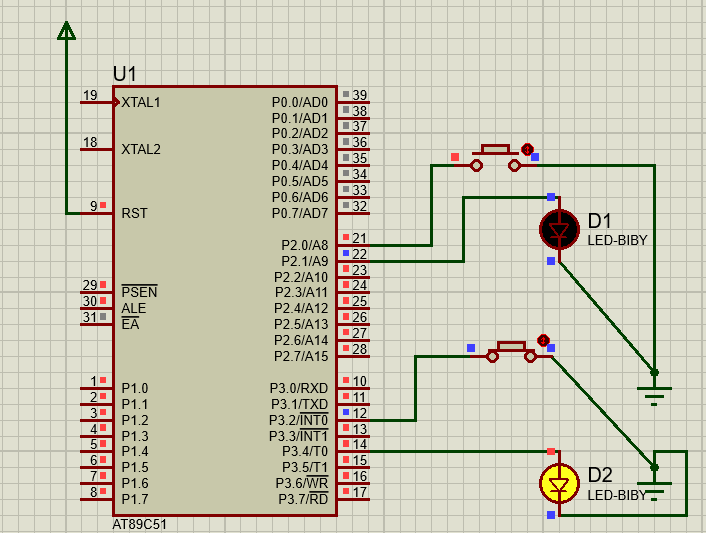


Fig 03: LED on port 3 blinks when button is pressed

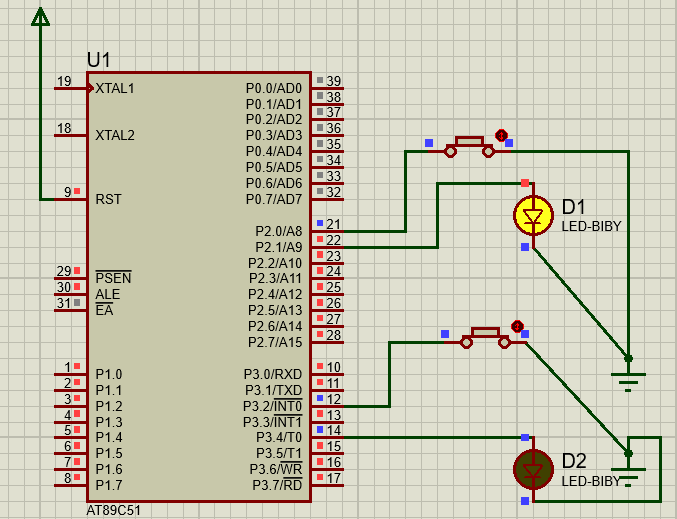


Fig 04: LED on port 2 light on pressing the button

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**2: Use Interrupts to count to F using 7 segment display.**

#include <reg51.h>

#include <stdio.h>

sbit I\_button = P3^2;

int count = 0;

void e\_i\_() interrupt 0

{

P1 = count++;

}

void main()

{

EA = 1;

EX0 = 1;

IT0 = 1;

while(1){}

}

C Code for counting to F in Hex on 8051 micro-controller

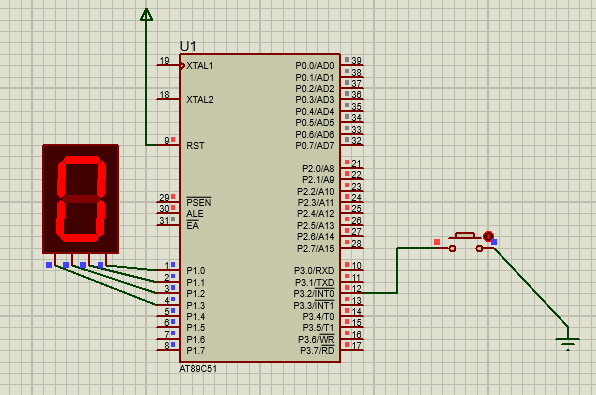


Fig 05: On pressing the 7-segment display increments

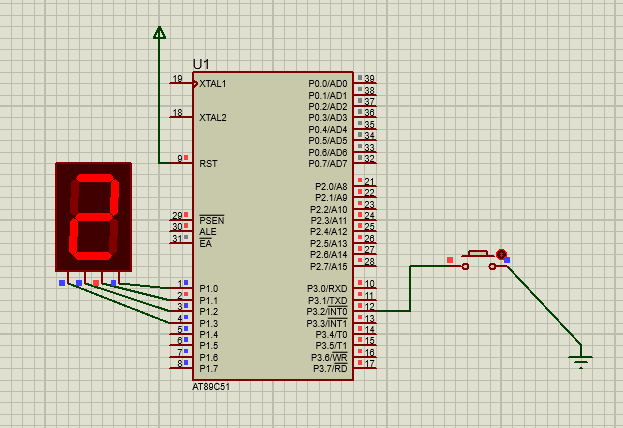
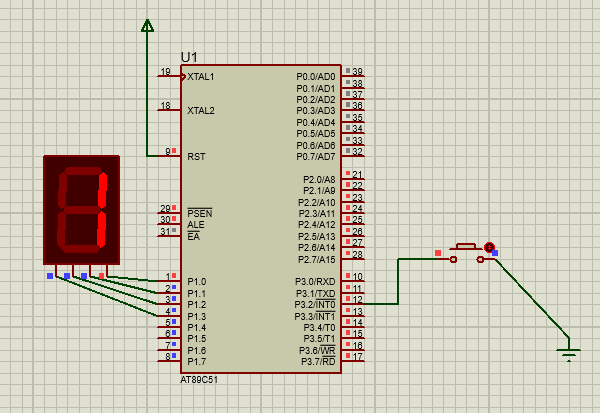


Fig 06: Increments to 1 and then 2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Conclusion:**

In this Lab we learned about interrupts and how to use them. In polling we monitor a pin constantly in an infinite loop and thus wasting CPU cycles where as in interrupts we don’t waste that many cycles. We just push a button that triggers the interrupt, which handles some function such lighting up an LED.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**