EE2004D-MICROPROCESSORS AND MICRO CONTROLLERS

PROJECT REPORT

LED CHASER CIRCUIT

Submitted by

Cholleti Nithin,
B200283EE

Annapareddy Rushendr Reddy,
B200302EE

INTRODUCTION

Light chasers are decorative lights or LEDs arranged in different moving patterns which create a chasing light or running light kind of effect. These look very interesting and are surely eye catching and that's why these types of lighting arrangement have gained immense popularity in today's world. They are widely used in advertising boards to make them look attractive.

BASIC THEORY

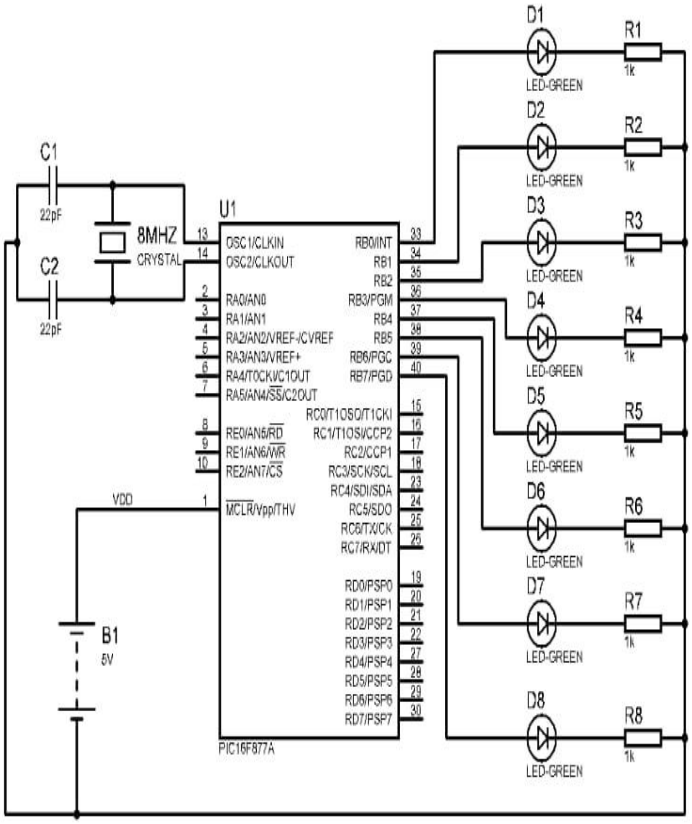
Her we are using PIC16F877A microcontroller to create the running. Time delay is the key feature used here. By controlling the output port and time delay we can create this circuit. There can numerous patterns created using this here we are going to do the following pattern

1. They get light one after the other in forward direction
2. They get light one after the other in backward direction
3. Then single light is off and moves forward
4. Then single light is off and moves backward
5. All lights turns on at a time and then goes off.

COMPONENTS REQUIRED

1. PIC16F877A microcontroller
2. Crystal oscillator (8MHz)
3. LED (general purpose)
4. Resistors
5. Capacitors
6. Batteries
7. Connecting Wires
8. Bread Board

SCHEMATIC DIAGRAM



COMPONENTS DESCRIPTION :

MICROCONTROLLER

Here we are using PIC16F877A which a 40 pin Microcontroller designed using RISC architecture and is widely used in embedded systems. We are using timers here for creating the delay it has three timers in it two of which are 8 bit timers and 1 is a 16 bit timer.

CRYSTAL OSCILLATOR

Crystal oscillators are mainly used in digital integrated circuits for providing a stable clock signal and in specific applications which require high frequency reference.

LED

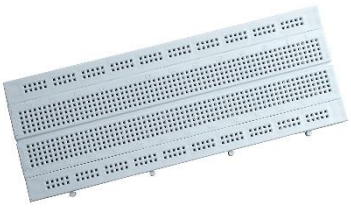
LED which is abbreviated as light emitting diode is a semiconductor light source that emits light when current flows through it .

BATTERY

A 6V dc voltage battery is used in the circuit to provide the required voltage

BREAD BOARD

A breadboard is a construction base for prototyping of electronic circuits as the bread board is solder less it is reusable which makes it easy for creating temporary prototypes.



CODE

```
#pragma config FOSC = XT
#pragma config WDTE = OFF
#pragma config PWRTE = OFF
#pragma config BOREN = OFF
#pragma config LVP = ON
#pragma config CPD = ON
#pragma config WRT = 256
#pragma config CP = ON



#include <stdio.h>
#include <stdlib.h>
#include <xc.h>

#define _XTAL_FREQ 8000000;

void Delay_ms(unsigned int a){
    for(unsigned int i=0;i<=a;i++){
                }
}
```

```
void main()
 {
    unsigned char i = 1, j = 128, k;
        TRISB = 0x00;
        PORTB = 0x00;

  while(1)
   {
        PORTB = 0xFF;
        Delay_ms(1000);

        for(k = 0; k <8; k ++)
      {
       PORTB = i;
       i = i<< 1;
       Delay_ms(200);
       if(i == 0){ i = 1;}
      }
     // right to left shift
     for(k = 0; k <8; k ++)
      {
        PORTB = j;
         j = j>>1;
          Delay_ms(200);
          if(j == 0){ j = 128;}
```

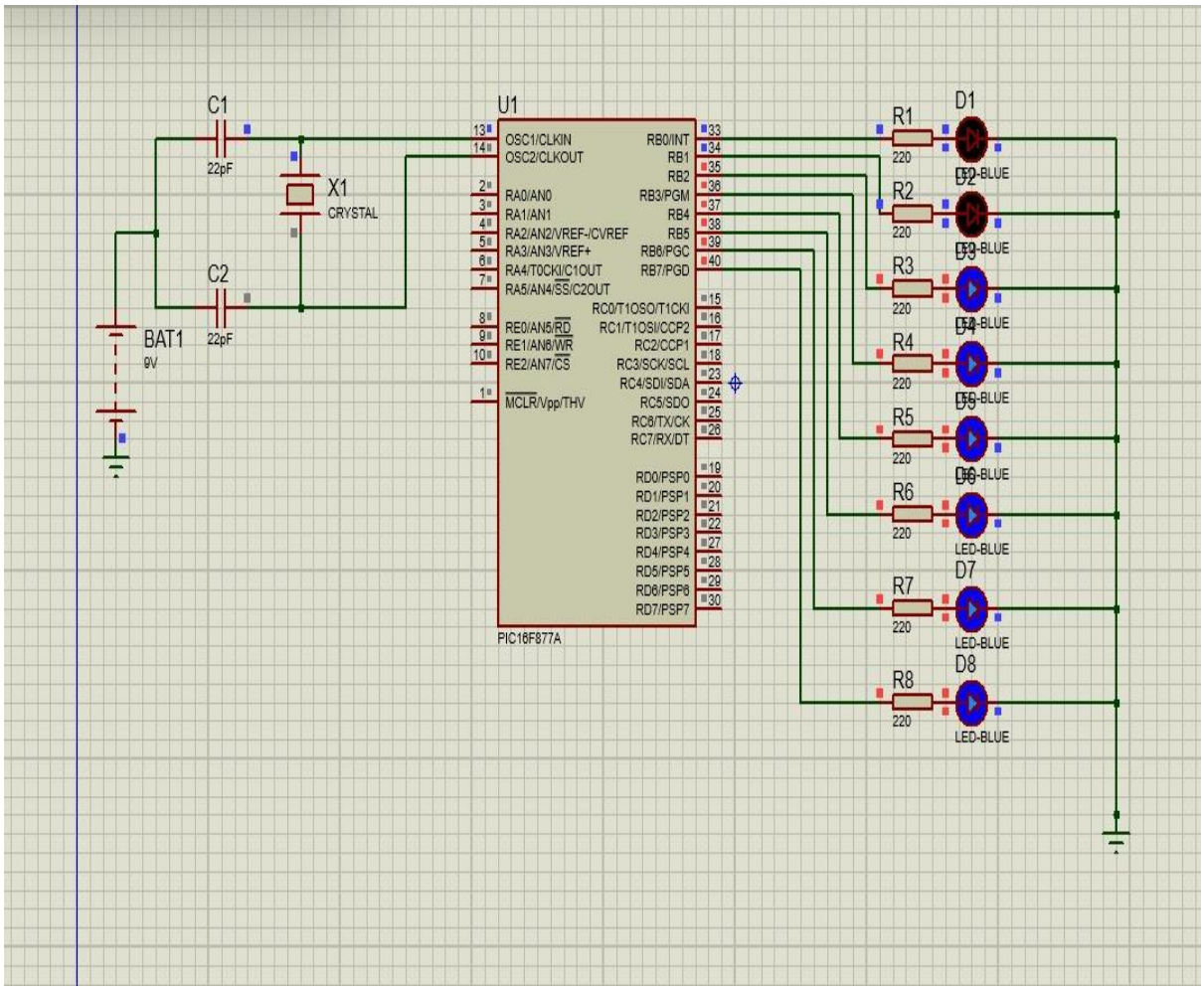
```
        }

       for(k = 0; k <8; k ++)
       {
        PORTB = i;
        i = i<<1 ;
        Delay_ms(200);
        if(i == 0){ i = 255;}
       }
       for(k = 0; k <8; k ++)
       {
         PORTB = j;
          j = j>>1;
          Delay_ms(200);
          if(j == 0){ j = 255;}
        }

    }
}
```

Code explanation:

Since PIC16 has no internal clock or oscillator, an external oscillator bit is enabled in XT configuration and disabling watchdog timer provides disabling of MCLR pin which means the microcontroller resets and sets automatically as the program progresses.

First we defined a function Delay_ms to create desired delay in the circuit then in the main function we took port B as the output port and made all its bits low. Now we make first pin of PORTB high and using the bit wise operator left shift(a<<b) we make the next pin of the port high and all others low as the pins are connected to LEDs this creates a racing effect in the forward direction. Similarly when the last pin of PORTB is high we use the bitwise operator right shift (a>>b) to create similar racing effect in the backward direction. After two cycles of this process port B is loaded with FFH which makes every bit high and every LED glows.

Output


C1
22pF
X1
CRYSTAL
C2
22pF
BAT1
9V
U1
OSC1/CLKIN
OSC2/CLKOUT
RA0/AN0
RA1/AN1
RA2/AN2/VREF-/CVREF
RA3/AN3/VREF+
RA4/T0CKI/C1OUT
RA5/AN4/SS/C2OUT
RE0/AN5/RD
RE1/AN6/WR
RE2/AN7/CS
MCLR/Vpp/THV
RB0/INT
RB1
RB2
RB3/PGM
RB4
RB5
RB6/PGC
RB7/PGD
RC0/T1OSO/T1CKI
RC1/T1OSI/CCP2
RC2/CCP1
RC3/SCK/SCL
RC4/SDI/SDA
RC5/SDO
RC6/TX/CK
RC7/RX/DT
RD0/PSP0
RD1/PSP1
RD2/PSP2
RD3/PSP3
RD4/PSP4
RD5/PSP5
RD6/PSP6
RD7/PSP7
PIC16F877A
R1
220
D1
R2
220
R3
220
R4
220
R5
220
R6
220
LED-BLUE
R7
220
D7
LED-BLUE
R8
220
D8
LED-BLUE