**Lab Task # 01**

Q: Write a program to blink built-in LED (Pin No.13, PB5) on UNO board at a frequency of 4 Hz with 50 % duty cycle?

**Code:**

#define LED 5

void setup( )

{

DDRB |= (1<<LED);

}

void loop( )

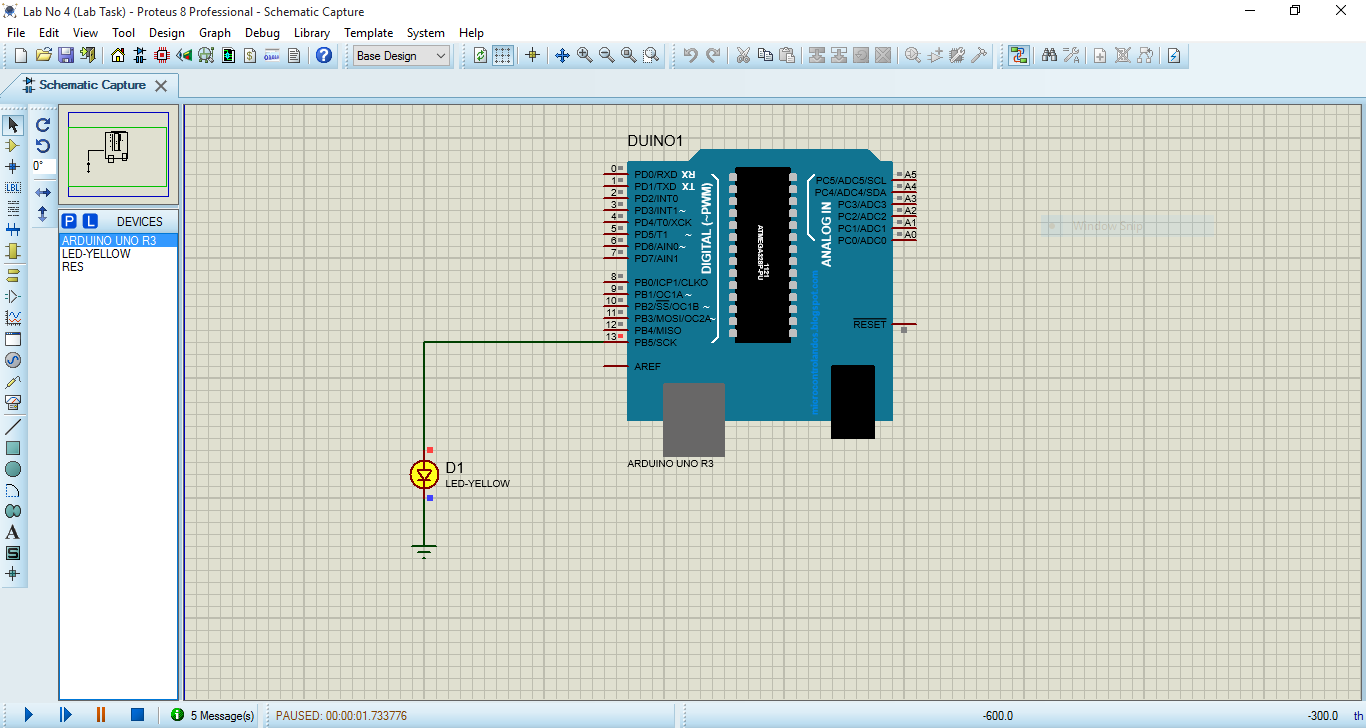
{

PORTB &= ~(1<<LED);

delay(500);

PORTB |= (1<<LED);

delay(500);

}

**Lab No#02**

Q1.: Show decimal numbers from 0 to 99 on two seven segment displays?

**Code:**

#define SEG0 8

#define SEG1 9

byte Count;

byte Seven\_Segment[] ={0x3F, 0x06, 0x5B, 0x4F, 0x66,0x6D, 0x7D, 0x07, 0x7F, 0x6F};

void Display(byte No)

{ byte units, tens;

tens = No / 10;

units = No % 10;

for (int I = 0 ; I < 20 ; I++)

{ digitalWrite(SEG1,LOW);

PORTD = Seven\_Segment[units];

digitalWrite(SEG0,HIGH);

delay(50);

digitalWrite(SEG0,LOW);

PORTD = Seven\_Segment[tens];

digitalWrite(SEG1,HIGH);

delay(50);

}

}

void setup()

{ DDRD = 0xFF;

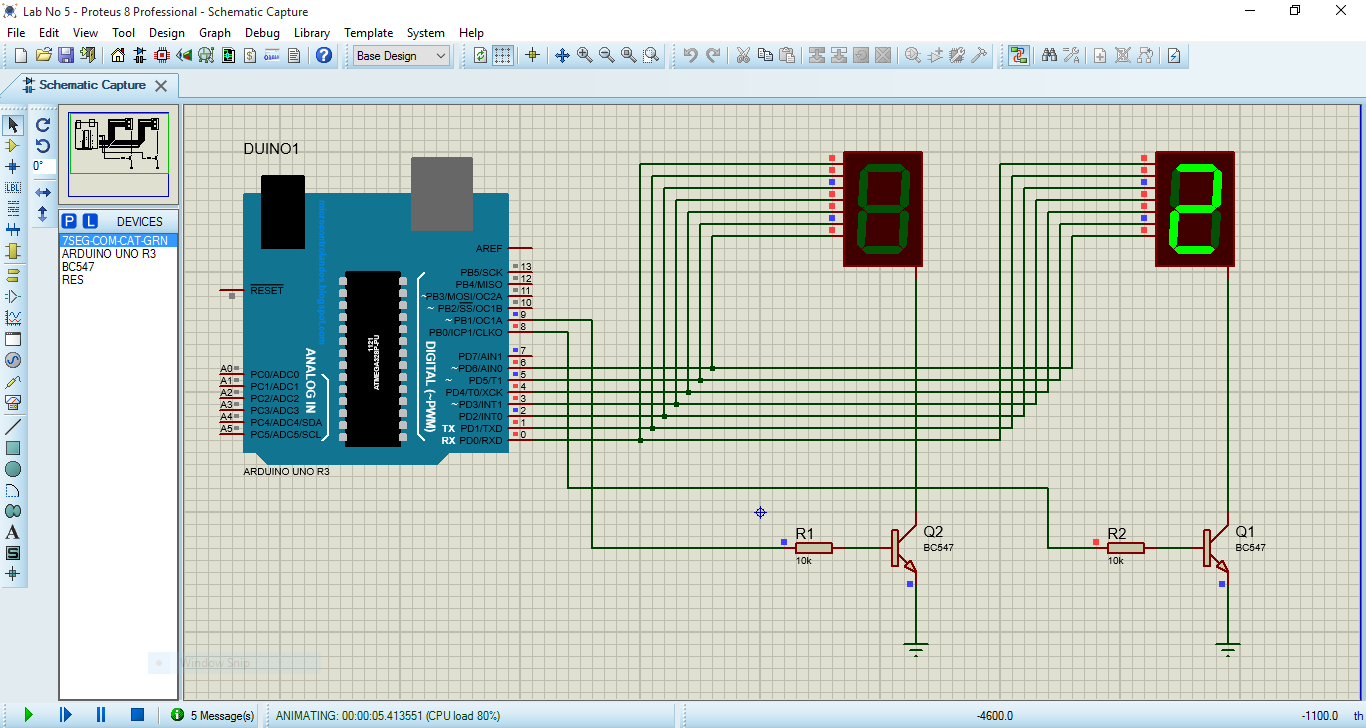
pinMode(SEG0,OUTPUT);

pinMode(SEG1,OUTPUT);}

void loop(){

Display(Count++);

Count = 0;}



Q: Write a sketch to display “Hello World” in first line and a character in second line of LCD?

**Code:**

#include <LiquidCrystal.h>

const int RS = 13, E = 12, D4 = 11, D5 = 10, D6 = 9, D7 = 8;

LiquidCrystal lcd(RS, E, D4, D5, D6, D7);

byte k=0;

byte Shape0[8]= {0b01110,0b01110,0b00100,0b01110,0b10101,0b00100,0b01010,0b01010};

byte Shape1[8]={ 0x0E,0x0E,0x15,0xE,0x04,0x04,0x0A,0x0A};

void setup()

{

lcd.begin(16, 2);

lcd.createChar(0, Shape0);

lcd.createChar(1, Shape1);

lcd.setCursor(0, 0);

lcd.print("hello, world!");

}

void loop()

{

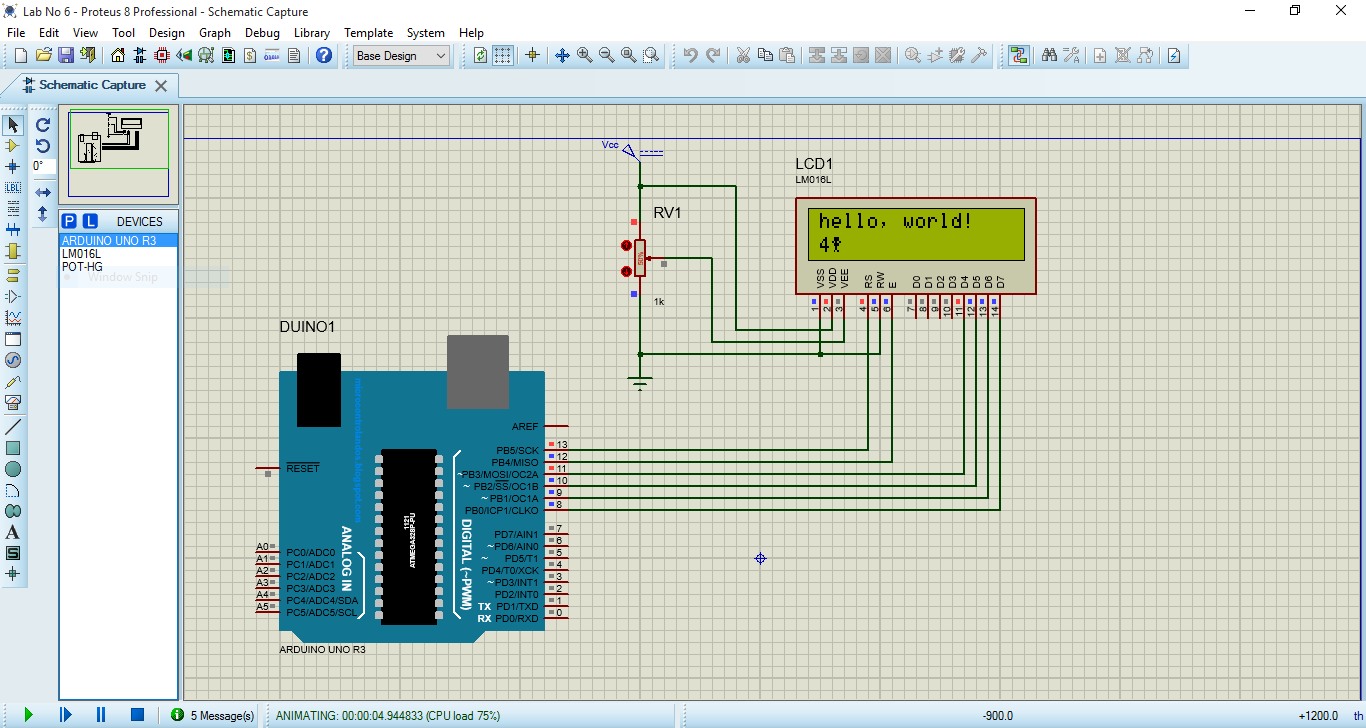
lcd.setCursor(0, 1);

lcd.print( millis() / 1000);

lcd.write(byte(k++%2));

delay(500);

}



Q: Write a sketch to display your registration number in first line LCD?

**Code:**

#include <LiquidCrystal.h>

const int RS = 13, E = 12, D4 = 11, D5 = 10, D6 = 9, D7 = 8;

LiquidCrystal lcd(RS, E, D4, D5, D6, D7);

byte k=0;

byte Shape0[8]={0x17,0x14,0x17,0x18,0x00,0x00,0x00,0x00};

byte Shape1[8]={0x00,0x00,0x00,0x13,0x12,0x11,0x1F,0x00};

void setup(){

lcd.begin(16,2);

lcd.clear();

lcd.createChar(0,Shape0);

lcd.createChar(1,Shape1);

lcd.setCursor(0,0);

lcd.print("Regd No: 21"); }

void loop()

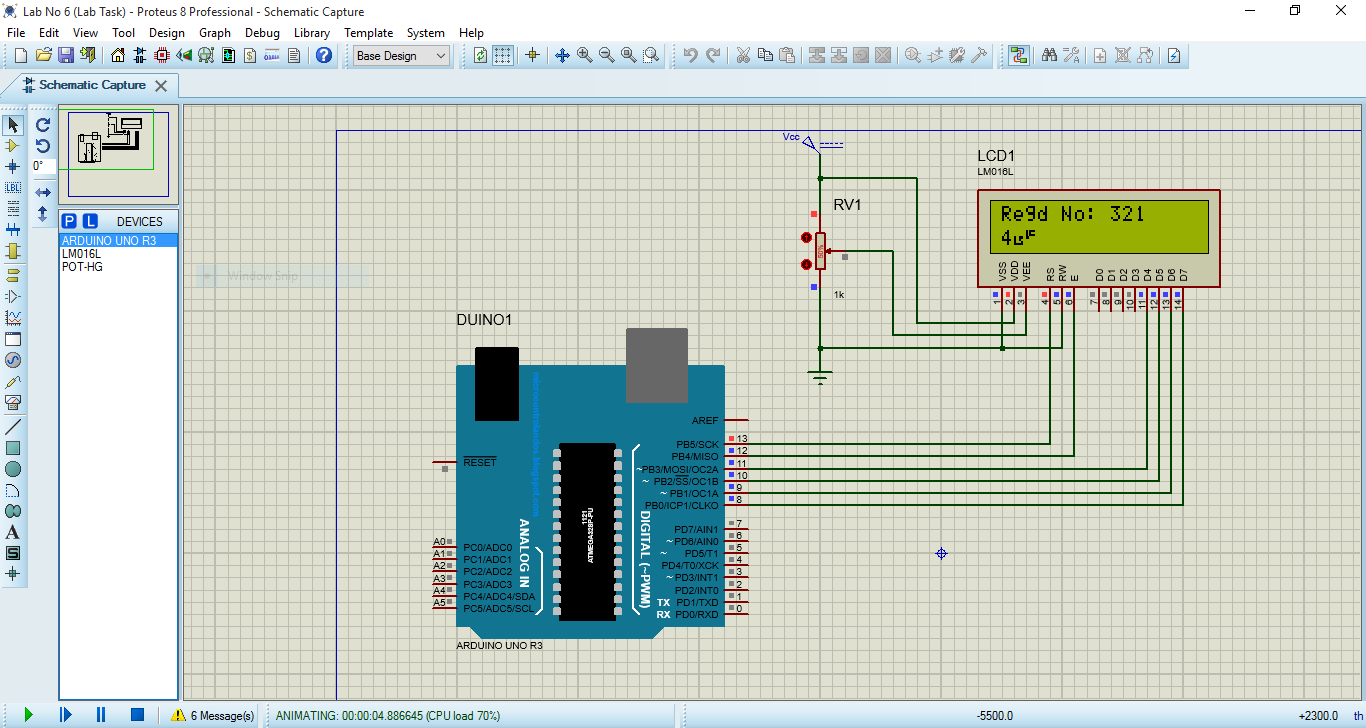
{ lcd.setCursor(0,1);

lcd.print( millis() / 1000);

lcd.write(byte(1));

lcd.write(byte(0));

delay(500);}



**LAB 03**

**Code:**

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <Keypad.h>

#include <DHT.h>

// Define DHT11 sensor

#define DHTPIN 5

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

// Define PIR sensor

#define PIRPIN 11

// Initialize the LCD

LiquidCrystal\_I2C lcd(0x27, 16, 2); // Adjust the address (0x27) if needed

// Define the 4x3 keypad

const byte ROWS = 4;

const byte COLS = 4;

char keys[ROWS][COLS] = {

{'1', '2', '3',},

{'4', '5', '6',},

{'7', '8', '9',},

{'\*', '0', '#',}

};

byte rowPins[ROWS] = {9, 8, 7, 6}; // Connect to the row pins of the keypad

byte colPins[COLS] = {4, 3, 2}; // Connect to the column pins of the keypad

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

void setup() {

// Initialize LCD

lcd.begin();

lcd.backlight();

lcd.print("System Initializing");

// Initialize DHT sensor

dht.begin();

// Initialize PIR sensor

pinMode(PIRPIN, INPUT);

// Wait for 2 seconds

delay(2000);

lcd.clear();

lcd.print("Ready");

delay(1000);

}

void loop() {

// Read keypad input

char key = keypad.getKey();

if (key) {

lcd.clear();

if (key == '1') {

// Read PIR sensor

int pirState = digitalRead(PIRPIN);

if (pirState == HIGH) {

lcd.print("Motion: Detected");

} else {

lcd.print("Motion: None");

}

} else if (key == '2') {

// Read temperature and humidity from DHT11

float temperature = dht.readTemperature();

float humidity = dht.readHumidity();

// Check if the readings are valid

if (isnan(temperature) || isnan(humidity)) {

lcd.print("Sensor Error");

} else {

lcd.print("Temp: ");

lcd.print(temperature);

lcd.print("C");

lcd.setCursor(0, 1);

lcd.print("Humidity: ");

lcd.print(humidity);

lcd.print("%");

}

} else {

lcd.print("Invalid Key");

}

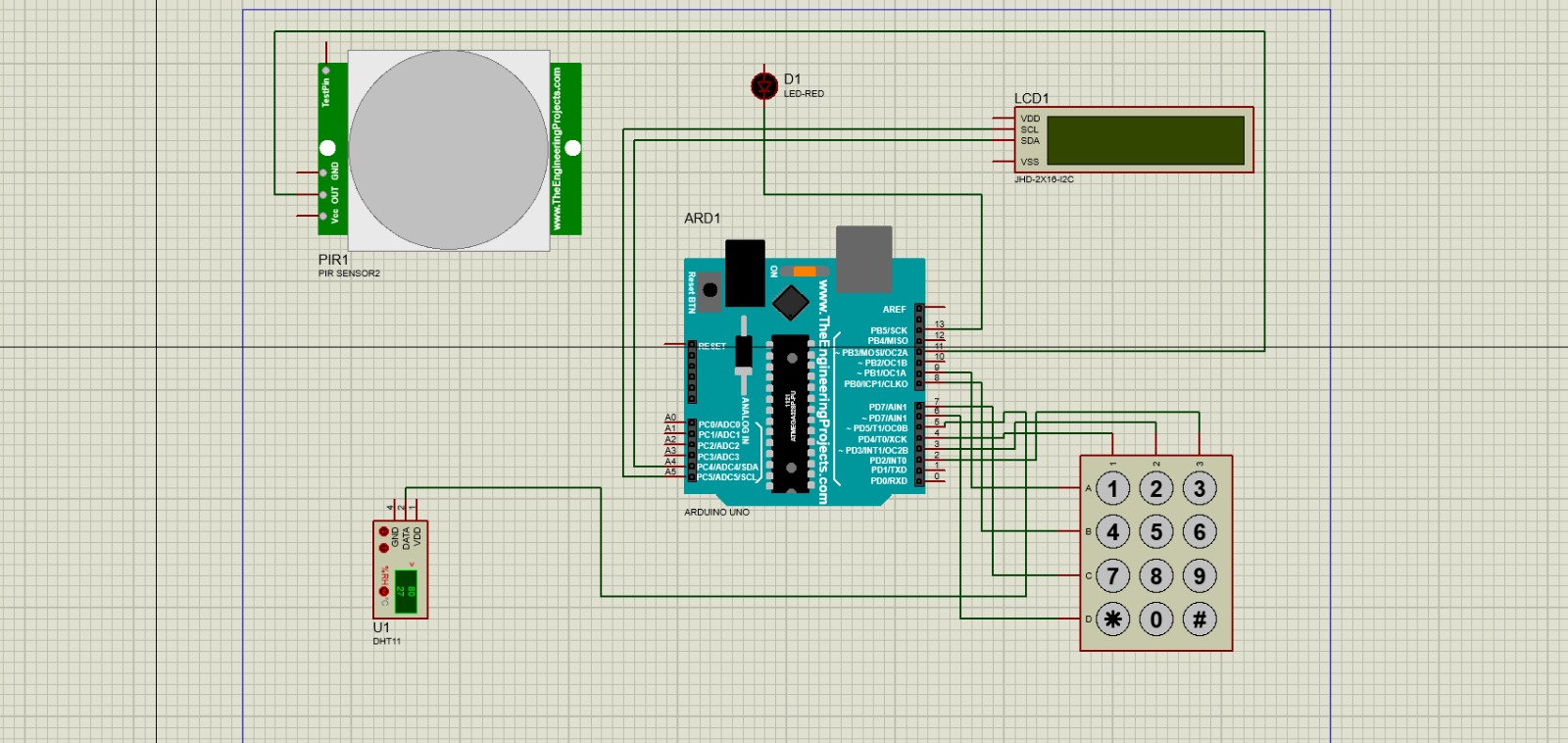
delay(2000);

lcd.clear();

}

}

**Schematics:**



**Lab No 4**

Q: Using Timer0 write a program to generate a Frequency 39.06 Hz on PB5 using Normal Mode or CTC Mode

**Code:**

const int outputPin = 13; // Built-in LED pin (PB5)

const int inputPin = 2; // Input pin for frequency measurement

volatile unsigned long lastRisingEdge = 0;

volatile unsigned long period = 0;

volatile bool newMeasurement = false;

void setup() {

// Initialize serial communication

Serial.begin(9600);

// Configure Timer0 for CTC mode

pinMode(outputPin, OUTPUT);

TCCR0A = 0;

TCCR0B = 0;

TCNT0 = 0;

// Set CTC mode and toggle OC0A on Compare Match

TCCR0A |= (1 << WGM01) | (1 << COM0A0);

// Set prescaler to 1024

TCCR0B |= (1 << CS02) | (1 << CS00);

// Set OCR0A for 39.06 Hz

OCR0A = 200;

// Configure input pin for measurement

pinMode(inputPin, INPUT);

attachInterrupt(digitalPinToInterrupt(inputPin), measurePeriod, RISING);

// Display initial terminal header

displayTerminalHeader();

}

void loop() {

static unsigned long lastDisplay = 0;

const unsigned long displayInterval = 1000; // Update every second

if (millis() - lastDisplay >= displayInterval) {

displayMeasurements();

lastDisplay = millis();

}

}

void measurePeriod() {

unsigned long currentTime = micros();

period = currentTime - lastRisingEdge;

lastRisingEdge = currentTime;

newMeasurement = true;

}

void displayTerminalHeader() {

Serial.println("\n$ Starting frequency measurement...");

Serial.println("============================");

Serial.println("Frequency Measurement Results:");

Serial.println("----------------------------");

}

void displayMeasurements() {

// Clear previous lines (ANSI escape codes)

Serial.println("\033[2J\033[H"); // Clear screen and move cursor to home

displayTerminalHeader();

// Calculate and display frequency

float freq = 1000000.0 / period; // Convert microseconds to Hz

float timePeriod = period / 1000.0; // Convert to milliseconds

// Format and display measurements

Serial.print("Measured Frequency: ");

Serial.print(freq, 2); // Display with 2 decimal places

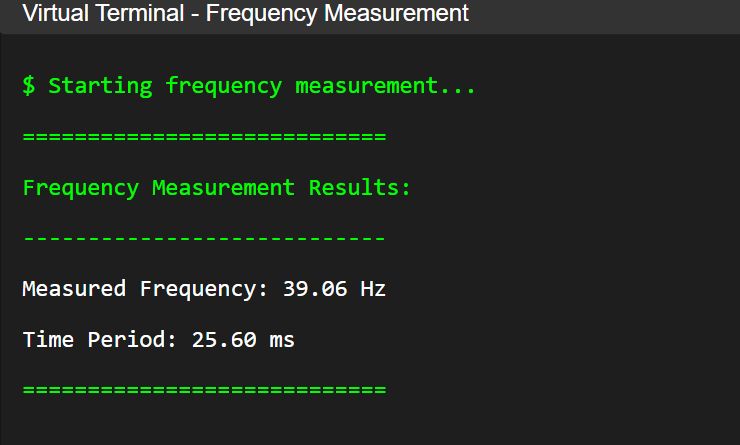
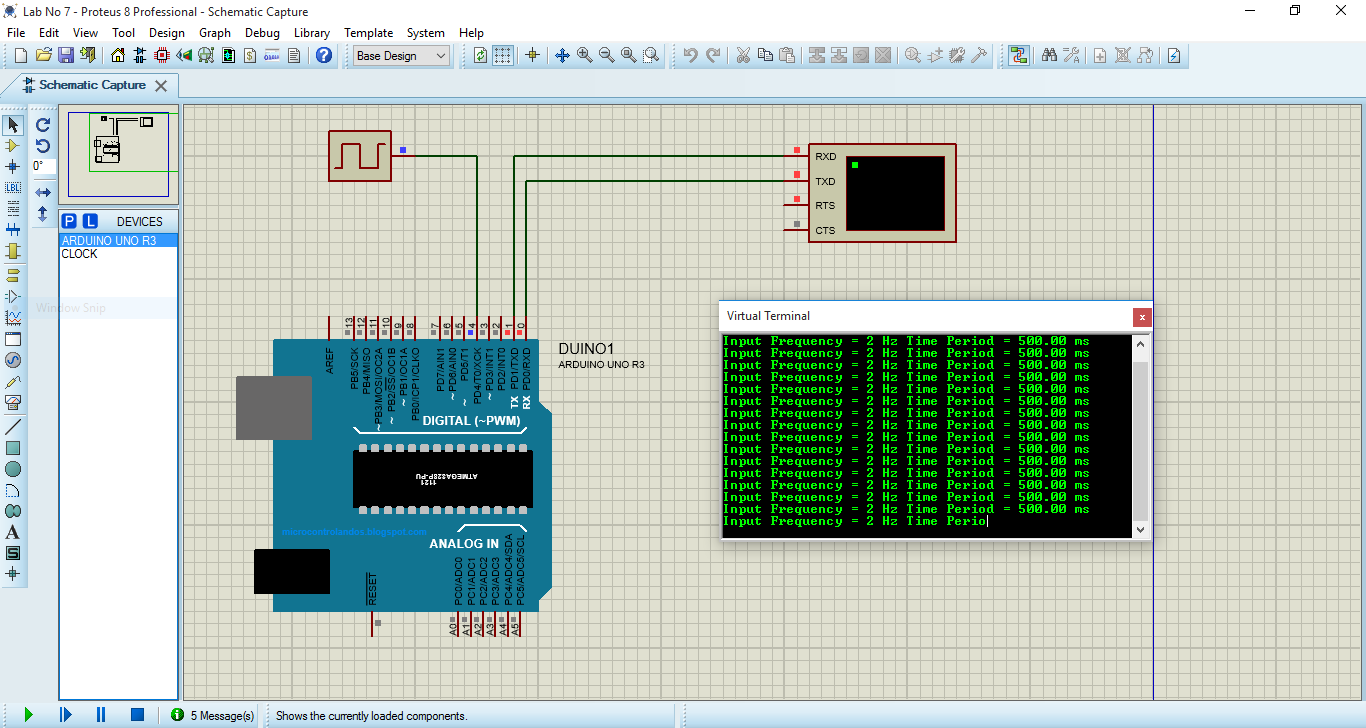
Serial.println(" Hz");

Serial.print("Time Period: ");

Serial.print(timePeriod, 2); // Display with 2 decimal places

Serial.println(" ms");

Serial.println("============================");

}

Q: Heart pulses of a patient, in the form of square wave are reaching at Pin T0 (PD4) (Arduino PIN 4) of Arduino UNO Board. Write a program to measure the current pulse rate per minute of that patient after each 20 seconds and send this answer via serial port to Computer?

**Code:**

void T1\_Delay()

{

TCNT1 = 0x0000;

OCR1A = 15625 - 1;

TCCR1A = 0x00;

TCCR1B = 0x0D;

while ((TIFR1 & (1<<OCF1A))==0);

TCCR1B = 0;

TIFR1 |= 1<<OCF1A;

}

void setup()

{

Serial.begin(9600);

DDRD&=~(1<<4);

PORTD|=(1<<4);

SREG&=(1<<7);

}

void loop()

{

TCNT0=0x00;

TCCR0A=0x00;

TCCR0B=0X06;

for(int i=0;i<20;i++)

{T1\_Delay();}

TCCR0B=0x00;

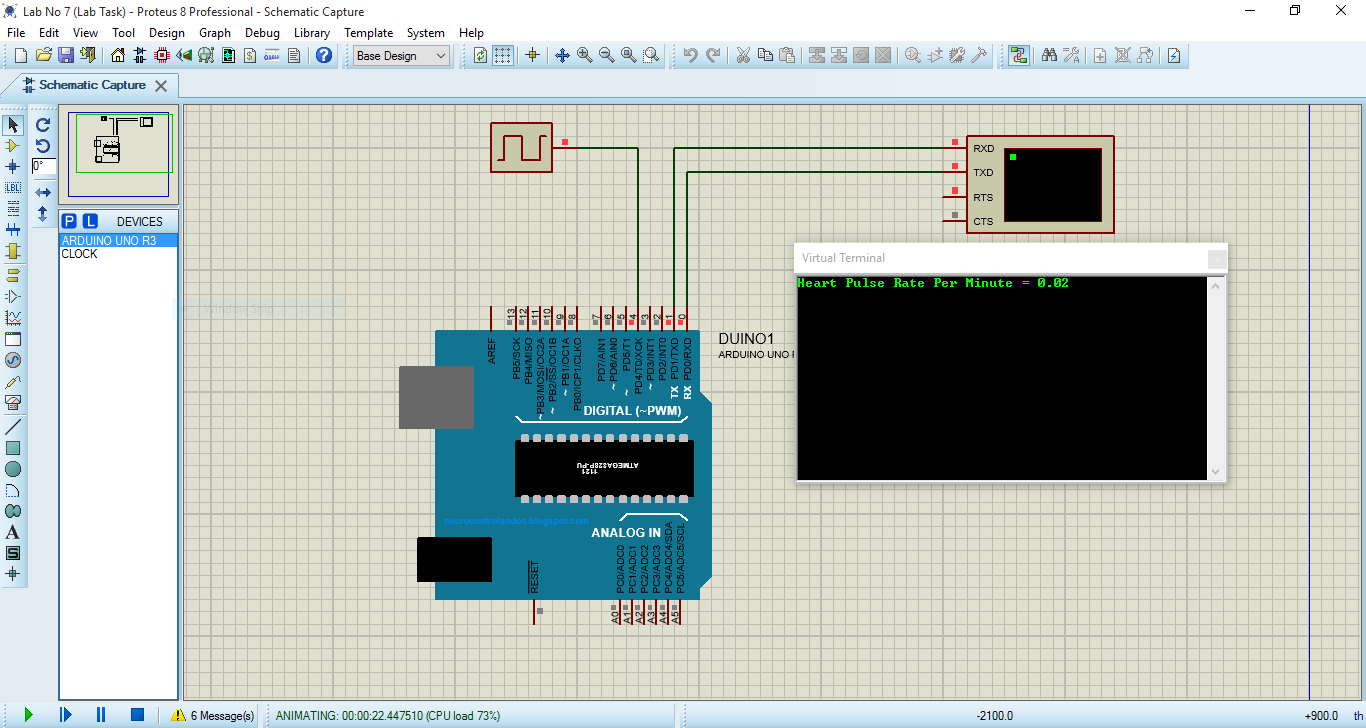
float Tp=1/TCNT1;

float Tp1=Tp/60;

Serial.print("Heart Pulse Rate Per Minute = ");

Serial.println(Tp1);

}



**Lab Task #06**

Q: Two IR sensors at a distance of 1 meter are placed on a road. Sensor0 is connected to INT0 interrupt pin and Sensor1 is connected to INT1 interrupt pin. A moving car crosses the Sensor0 first and then Sensor1. Calculate the time difference between two sensors detection and then the speed of that Car in Kilometer per Hour Units. Send these two answers via serial port to PC.

**Code:**

unsigned long Sensor0 = 0;

unsigned long Sensor1 = 0;

unsigned long Time;

float Time\_In\_Seconds;

float Distance = 1; // 1 Meter

float Speed;

float Speed\_In\_Km\_Per\_Hour;

void setup()

{

Serial.begin(9600);

DDRD = DDRD & 0b11110011;

PORTD = PORTD | 0b00001100;

EIMSK = EIMSK | 0b00000011;

EICRA = 0b00001010;

SREG = SREG | (1 << 7);

}

void loop()

{

}

ISR(INT0\_vect)

{

Sensor0 = millis();

}

ISR(INT1\_vect)

{

Sensor1 = millis();

Time = Sensor1 - Sensor0;

Time\_In\_Seconds = Time/1000;

Speed = Distance/Time\_In\_Seconds;

Speed\_In\_Km\_Per\_Hour = (Speed\*3600)/1000;

Serial.print(" Time Difference = ");

Serial.print(Time\_In\_Seconds);

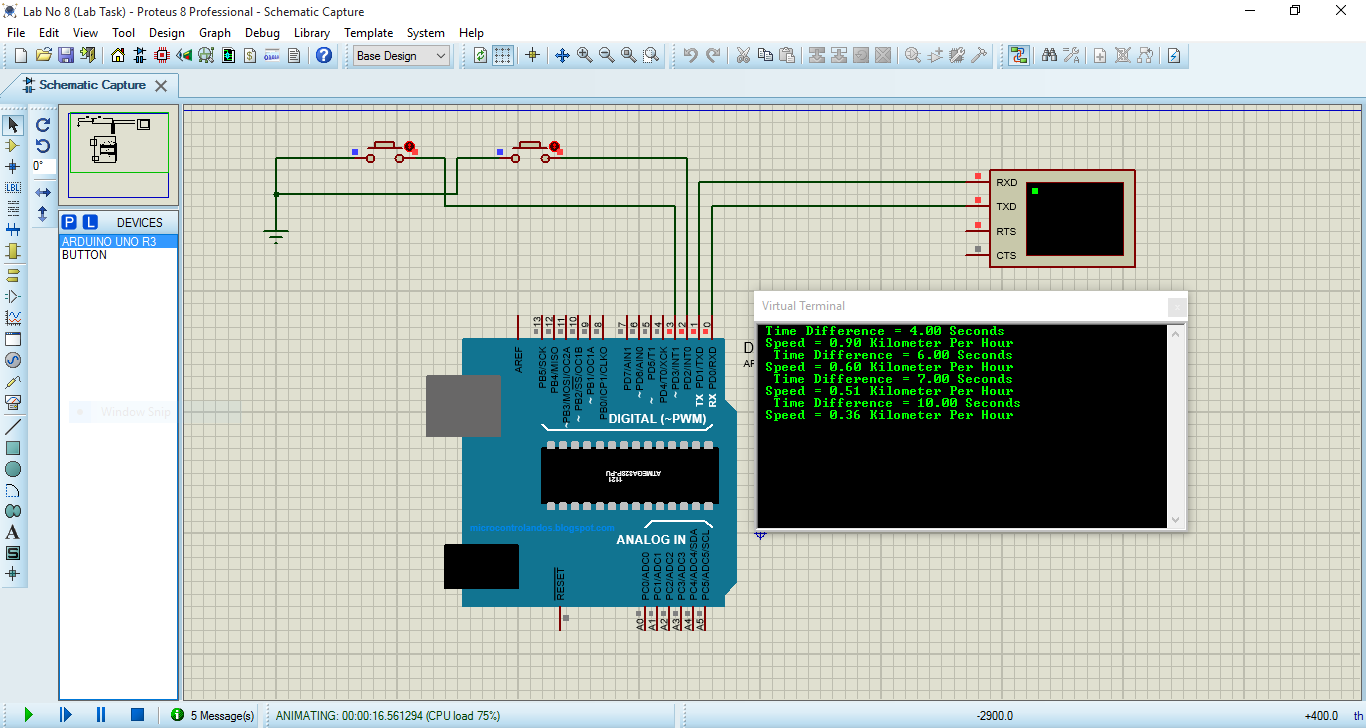
Serial.print(" Seconds\n\r ");

Serial.print("Speed = ");

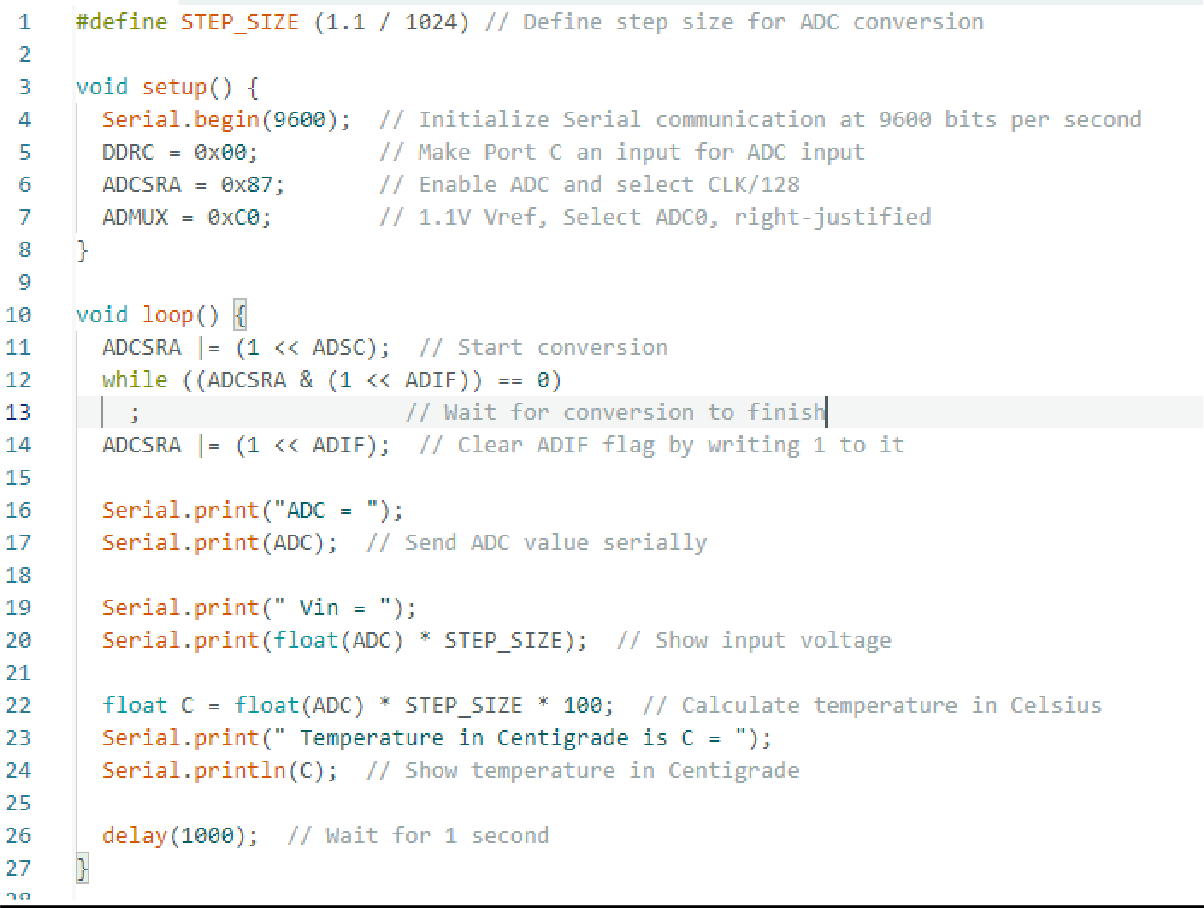
Serial.print(Speed\_In\_Km\_Per\_Hour);

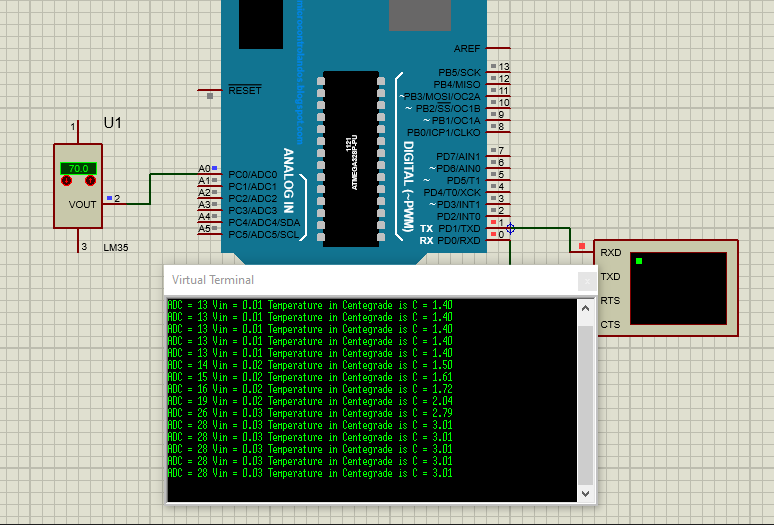
Serial.print(" Kilometer Per Hour\n\r ");

delay(200000);}



Task 7





**Lab No 9**

**Code:**

void setup()

{

DDRB = 0xFF;

PORTB = 0xFF;

Serial.begin(9600);

}

void loop()

{

if (Serial.available())

{

int inByte = Serial.read();

switch(inByte)

{

case '0': // if received byte is '1' = 0x31

PORTB |= (1<<5); // Turn ON LED

Serial.println("0 - Fan is OFF Now");

break;

case '1': // if received byte is '0' = 0x30

PORTB &= ~(1<<5); // Turn OFF LED

Serial.println("1 - Fan is ON Now");

break;

case '2': // if received byte is '2' = 0x32

if(PORTB &(1<<5)) Serial.println("2 - Fan Status = OFF");

else Serial.println("2 - Fan Status = ON");

break;

default: // if received byte is defferent

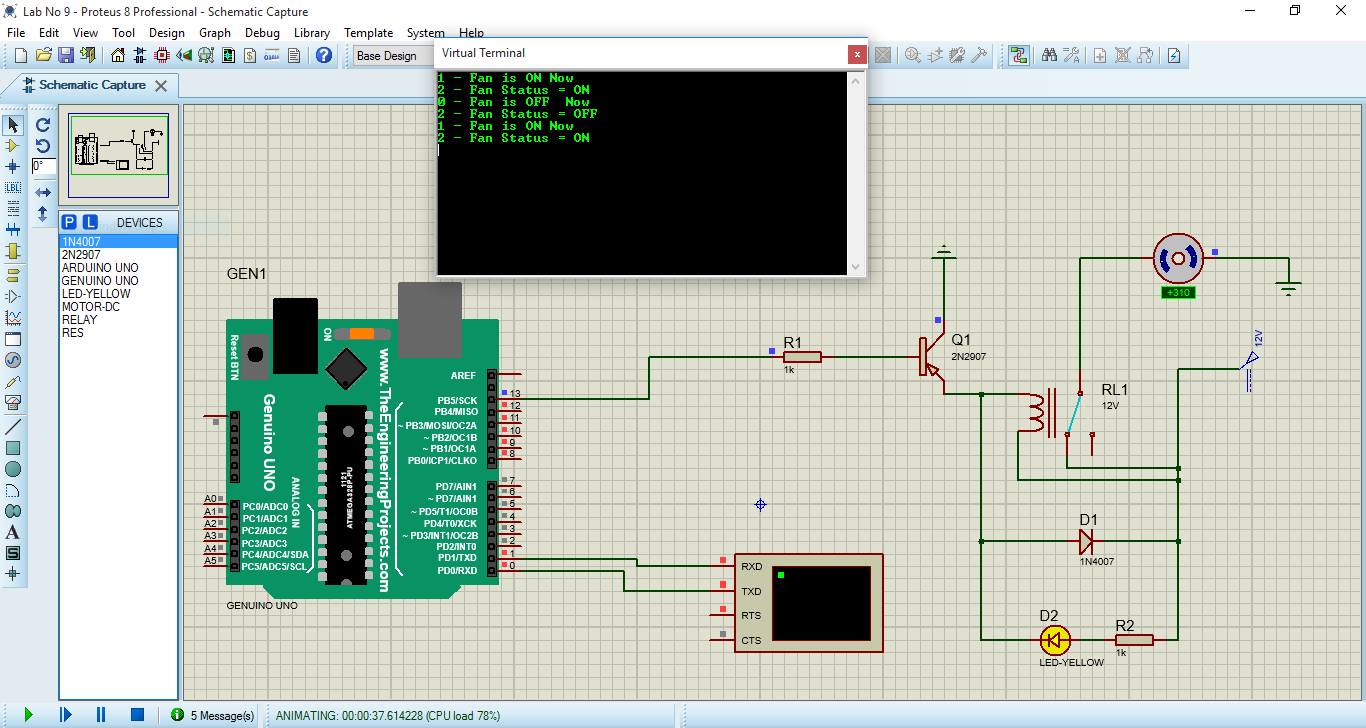
Serial.write(inByte);

Serial.println(" - is Unrecognized Command");

}

}

}



**Lab Task5**

Q: An LED is connected to Arduino Pin No.13(PB5). Write a Program that that receives a byte serially and acts according to following table?

|  |  |
| --- | --- |
| Received Byte | Action to Perform |
| '0' | Turn OFF LED and send back message |
| '1' | Turn ON LED and send back message |
| '2' | Turn ON LED 2 times with some delay and send back message |
| '3' | Turn ON LED 3 times with some delay and send back message |
| '4' | Turn ON LED 4 times with some delay and send back message |
| '9' | Send back message about the (ON/OFF) status of LED |
| Any other byte | Send back message that this byte is not a valid command |

**Code:**

#define FOSC 16000000UL // Clock Speed

#define BAUD 9600

#define MYUBRR FOSC/16/BAUD - 1

void USART\_Init()

{

UBRR0H = MYUBRR >> 8; // Set baud rate

UBRR0L = MYUBRR; // UBRR0L = 103

UCSR0B = (1<<RXEN0 )|(1<<TXEN0 ); // Enable receiver and transmit.

UCSR0C = (1<<UCSZ01)|(3<<UCSZ00); // 8 data bits, 1 stop bit

}

byte USART\_Receive()

{

while ( !(UCSR0A & (1<<RXC0)) ); // Wait for data to be received

return UDR0; // Return received data

}

void USART\_Send\_Str (byte data[])

{

for (int i = 0 ; data[i]; i++ )

{

while (! (UCSR0A & (1<<UDRE0))); // wait until UDR0 is empty

UDR0 = data[i]; // transmit data

}

}

void setup()

{

USART\_Init();

DDRB |= (1<<5);

}

void loop()

{

byte A = USART\_Receive();

switch (A)

{

case '0':

PORTB &= ~(1<<5);

USART\_Send\_Str ("LED Is OFF\n\r"); break;

case '1':

PORTB |= (1<<5);

USART\_Send\_Str ("LED Is ON\n\r"); break;

case '2':

USART\_Send\_Str ("Turn ON LED 2 Times With 2 Seconds Delay\n\r");

PORTB |= (1<<5);

delay(2000);

PORTB &= ~(1<<5);

delay(2000);

PORTB |= (1<<5);

delay(2000);

PORTB &= ~(1<<5);

delay(2000);

break;

case '3':

USART\_Send\_Str ("Turn ON LED 3 Times With 2 Seconds Delay\n\r");

PORTB |= (1<<5);

delay(2000);

PORTB &= ~(1<<5);

delay(2000);

PORTB |= (1<<5);

delay(2000);

PORTB &= ~(1<<5);

delay(2000);

PORTB |= (1<<5);

delay(2000);

PORTB &= ~(1<<5);

delay(2000);

break;

case '4':

USART\_Send\_Str ("Turn ON LED 4 Times With 2 Seconds Delay\n\r");

PORTB |= (1<<5);

delay(2000);

PORTB &= ~(1<<5);

delay(2000);

PORTB |= (1<<5);

delay(2000);

PORTB &= ~(1<<5);

delay(2000);

PORTB |= (1<<5);

delay(2000);

PORTB &= ~(1<<5);

delay(2000);

PORTB |= (1<<5);

delay(2000);

PORTB &= ~(1<<5);

delay(2000);

break;

case '9':

if ((PORTB &= (1<<5)))

{

USART\_Send\_Str ("LED Is ON\n\r");

}

else USART\_Send\_Str ("LED Is OFF\n\r");

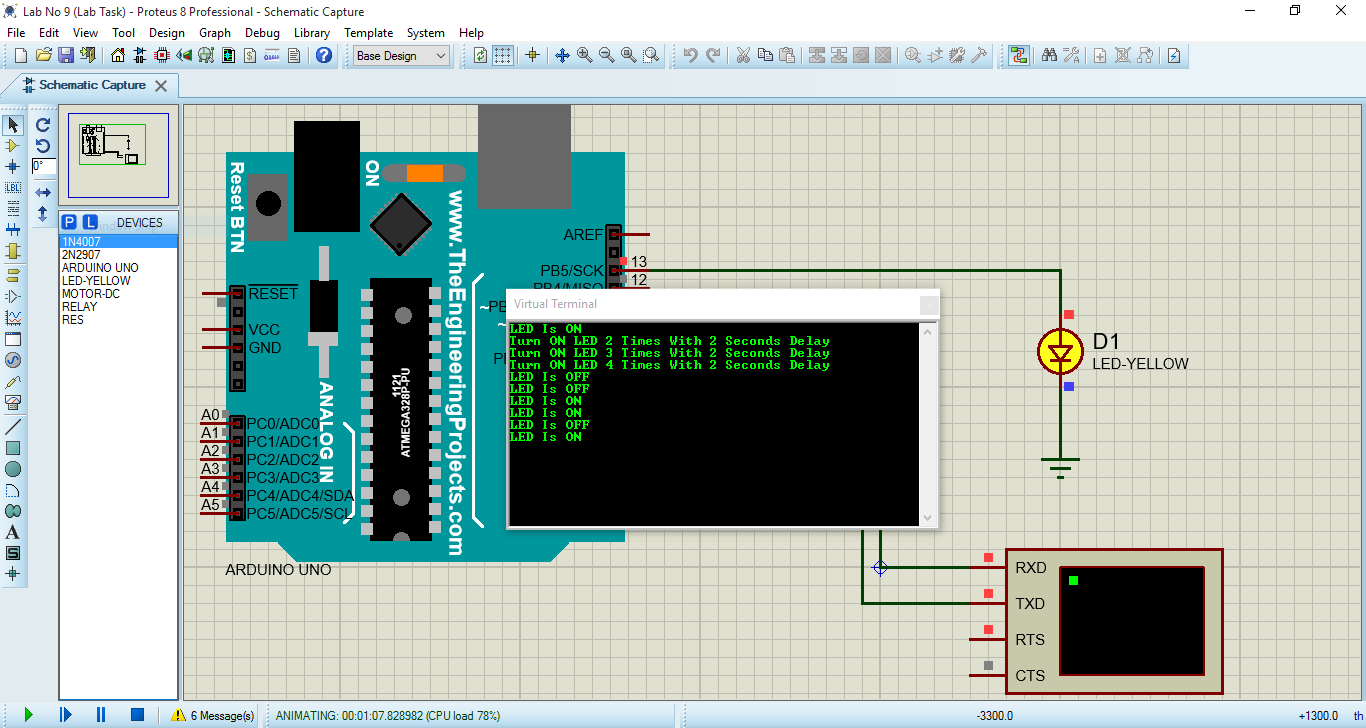
break;

default:

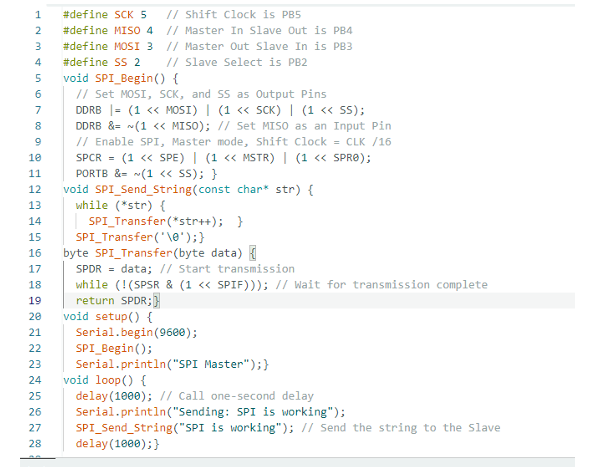
USART\_Send\_Str ("Error!!! Invalid Command.\n\r");

}

}



LAB 08



A screenshot of a computer program

Description automatically generated

A computer screen shot of a computer

Description automatically generated