**BÁO CÁO THỰC HÀNH**

**GIAO TIẾP ĐIỀU KHIỂN THIẾT BỊ NGOẠI VI**

*Tên bài thực hành:*

***THIẾT BỊ ĐẾM COUNTDOWN***

*Sinh viên thực hiện:*

* Bùi Mạnh Đạt – 18032741

Ngày thực hiện: 16 /06 /2021

**PHẦN I: Chuẩn đầu ra môn học : Trình bày được cấu tạo, nguyên lý hoạt động và biết sử dụng các thành phần trong hệ thống, thiết bị điện tử**

* 1. ***Mô tả tóm tắt nội dung bài thực hành***

**Yêu cầu:**

* Thiết bị sẽ đếm xuống từ giá trị cài đặt ban đầu (tối đa 99) về 00;
* Giá trị ban đầu này được cài đặt trên VS-C# (chỉ cho nhập số, tối đa 99, tối thiểu 9);
* Trên bảng điều khiển VS-C# có: 2 Led 7 đoạn hiển thị trên đó, 01 Button: START
* Khi nhấn START thì Led 7 đoạn trên bo và trên VS-C# cùng đếm xuống về 00, khi đã về 00 thì Led báo sáng 3 giây rồi tắt.
* Hiển thị giá trị đếm được lên màn hình LCD

**Nội dung:**

* Giao tiếp và điều khiển thiết bị ngoại vi thông qua Bluetooth giữa máy tính và vi điều khiển PIC18F4550
* Giao tiếp và điều khiển thiết bị ngoại vi thông qua Bluetooth giữa máy tính và Arduino Mega 2560
* Mô phỏng trên phần mềm Proteus
* Tạo giao diện điều khiển bằng Winform
* Xác định, liệt kê các linh kiện, thiết bị cần sử dụng.
* Vi điều khiển:
* PIC18F4550
* Adruino MEGA 2560
* Mạch giao tiếp USB – TTL : PL2303
* Linh kiện:
* Điện trở
* Led đơn
* Dây cắm
* Led 7 đoạn
  1. ***Sơ đồ khối của bài thực hành.***
* Vẽ sơ đồ khối.

Lled đơn

Lled 7 đoạn

LCD

Bluetooth

PIC 18F4550

hoặc

Adruino Mega 2560

PC, laptop….

Firmware

Hardware

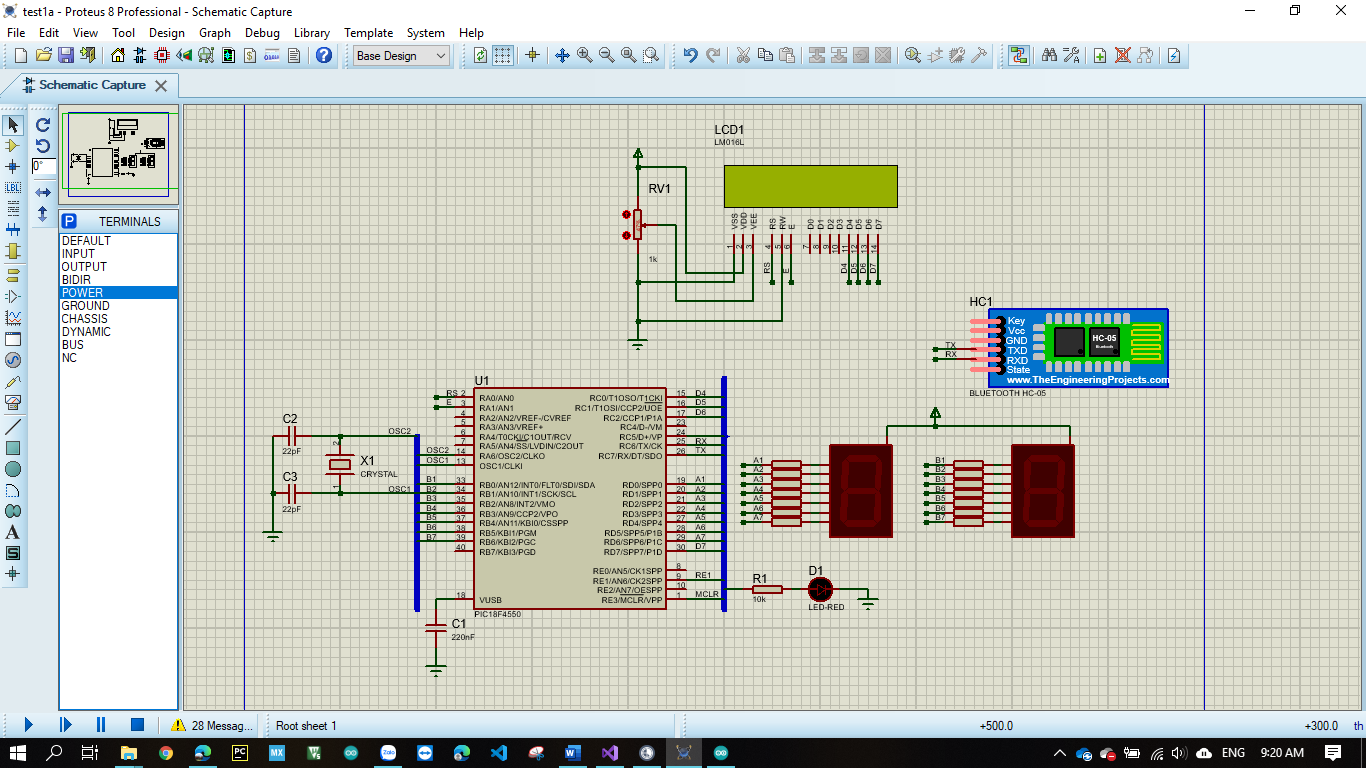
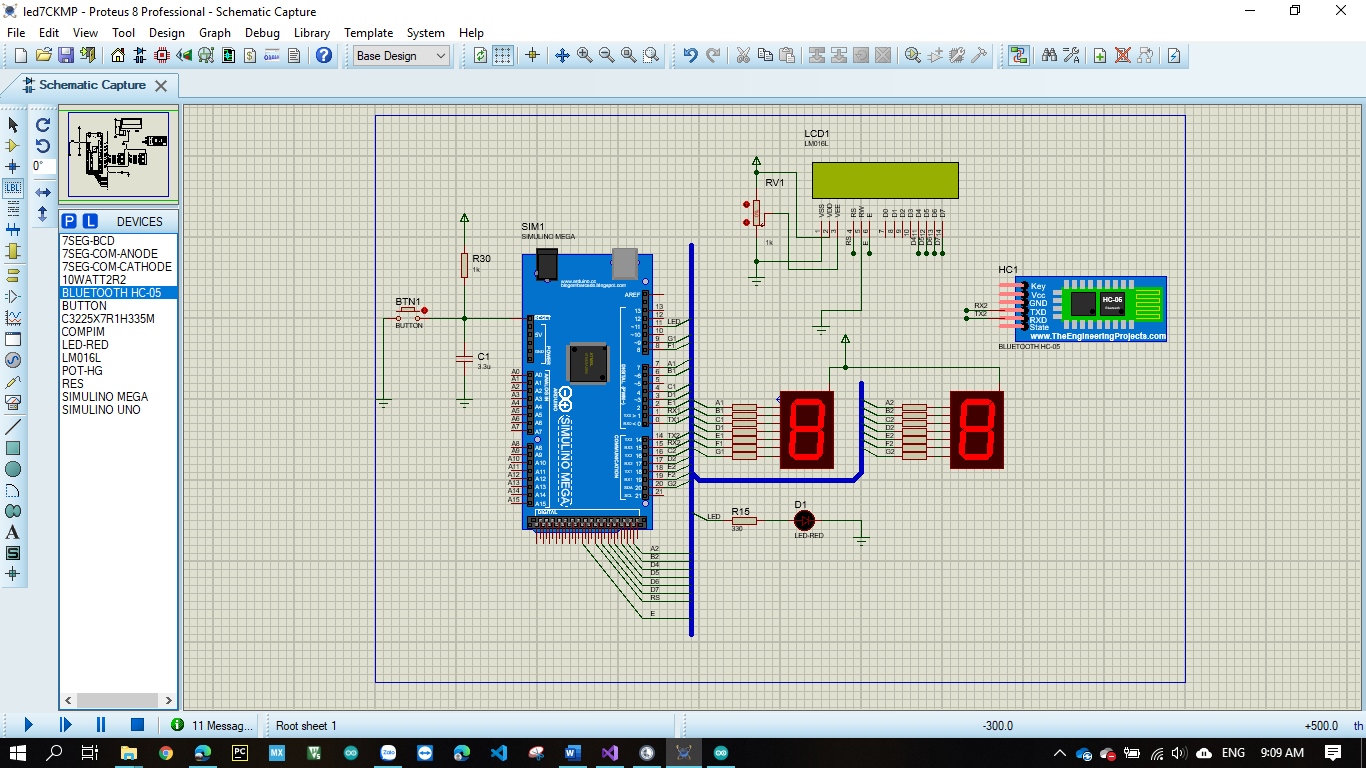
Software

* Giải thích chức năng, nhiệm vụ từng khối.
* Software :
* Bảng điều khiển:
* Hiển thị, điều khiển, gửi/ nhận dữ liệu giữa PC với vi điều khiển thông qua bluetooth
* Nhập giá trị bắt đầu, hiển thị giá trị đếm thông qua LED 7 đoạn
* Firmware:
* Chương trình điều khiển
* Hardware: Thiết bị ngoại vi
* Gồm đầu ra: led đơn, led 7 đoạn, LCD 16x2
* Đầu vào: Dữ liệu gửi từ bảng điều khiển
* Thực hiện các chức năng từ PC xuống vi điều khiển thông qua bluetooth và truyền tín hiệu về vi điều khiển để gửi lên PC

**PHẦN II: Chuẩn đầu ra môn học 2: Có khả sử dụng các phần mềm lập trình, mô phỏng**

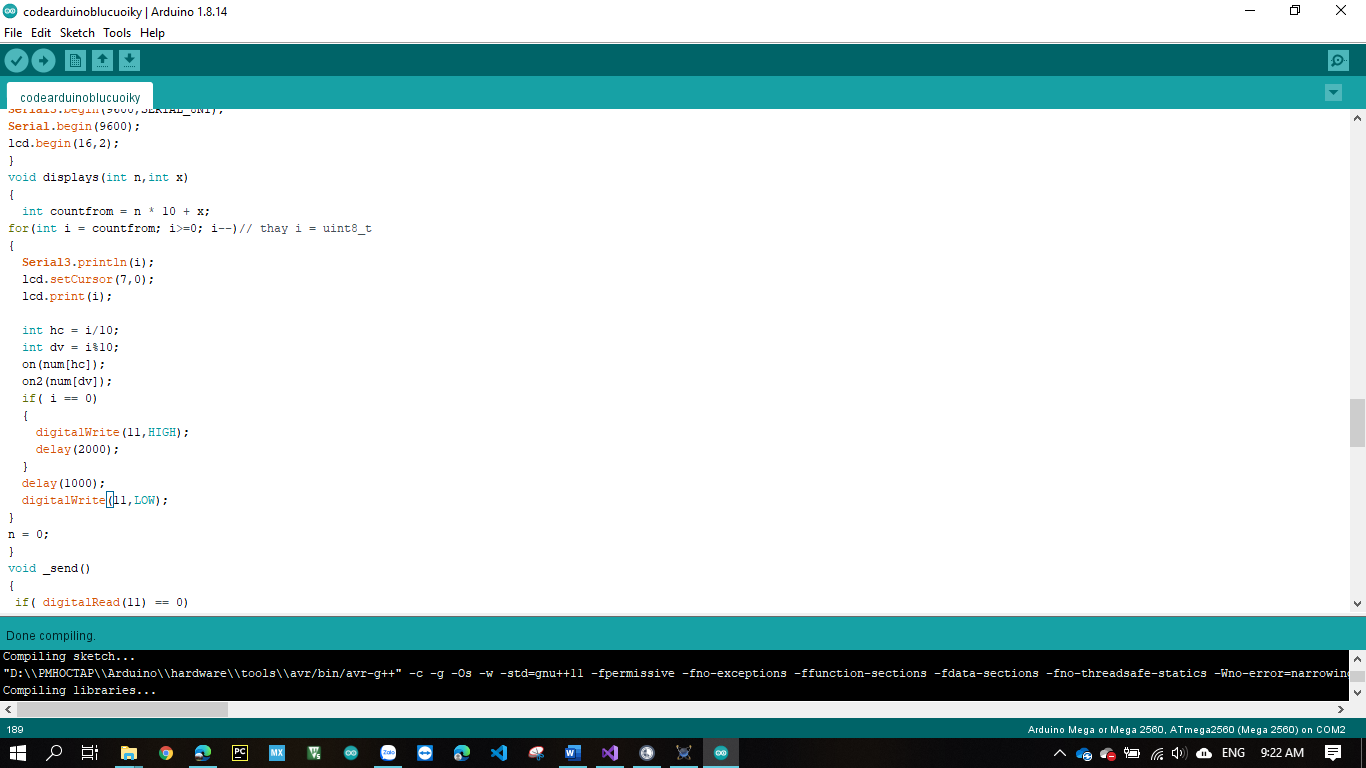
***2.1. Khả năng sử dụng phần mềm mô phỏng, vẽ mạch - Proteus***

Vẽ sơ đồ nguyên lý của mạch trong Proteus.

**Hình 2.1: Sơ đồ nguyên lý proteus với PIC**

**Hình 2.2 Sơ đồ nguyên lý với Arduino**

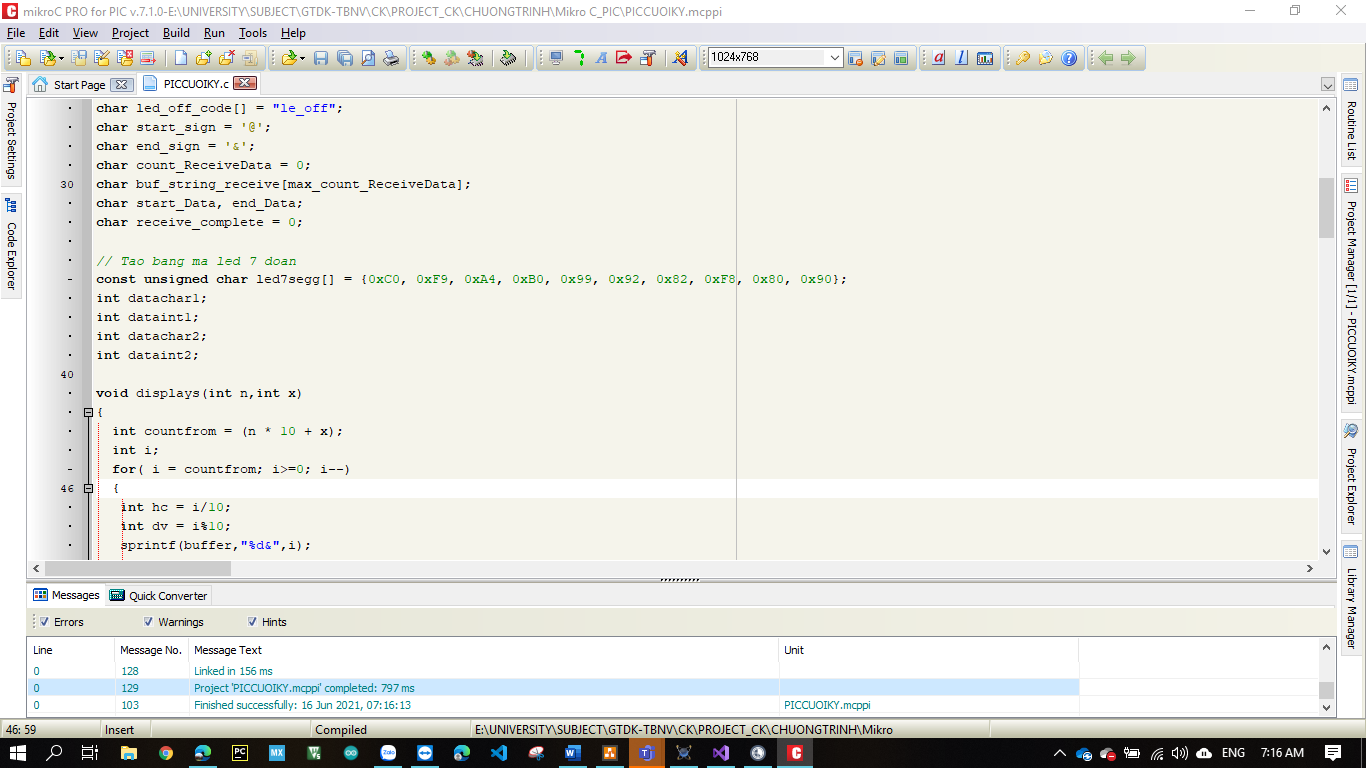
***2.2. Khả năng sử dụng phần mềm lập trình Arduino***



**Hình 2.3: Lập trình trên Arduino IDE**

***2.3. Khả năng sử dụng phần mềm lập trình cho PIC - MikroC***

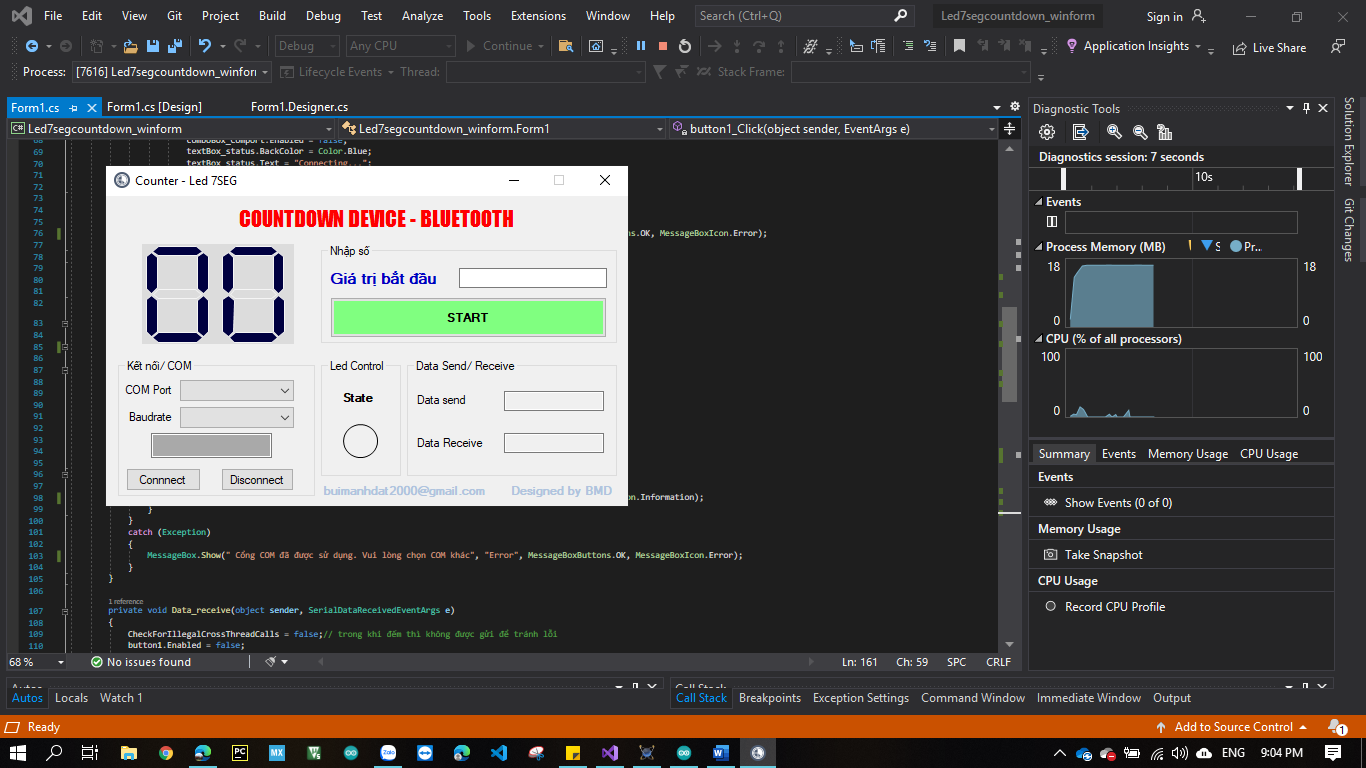
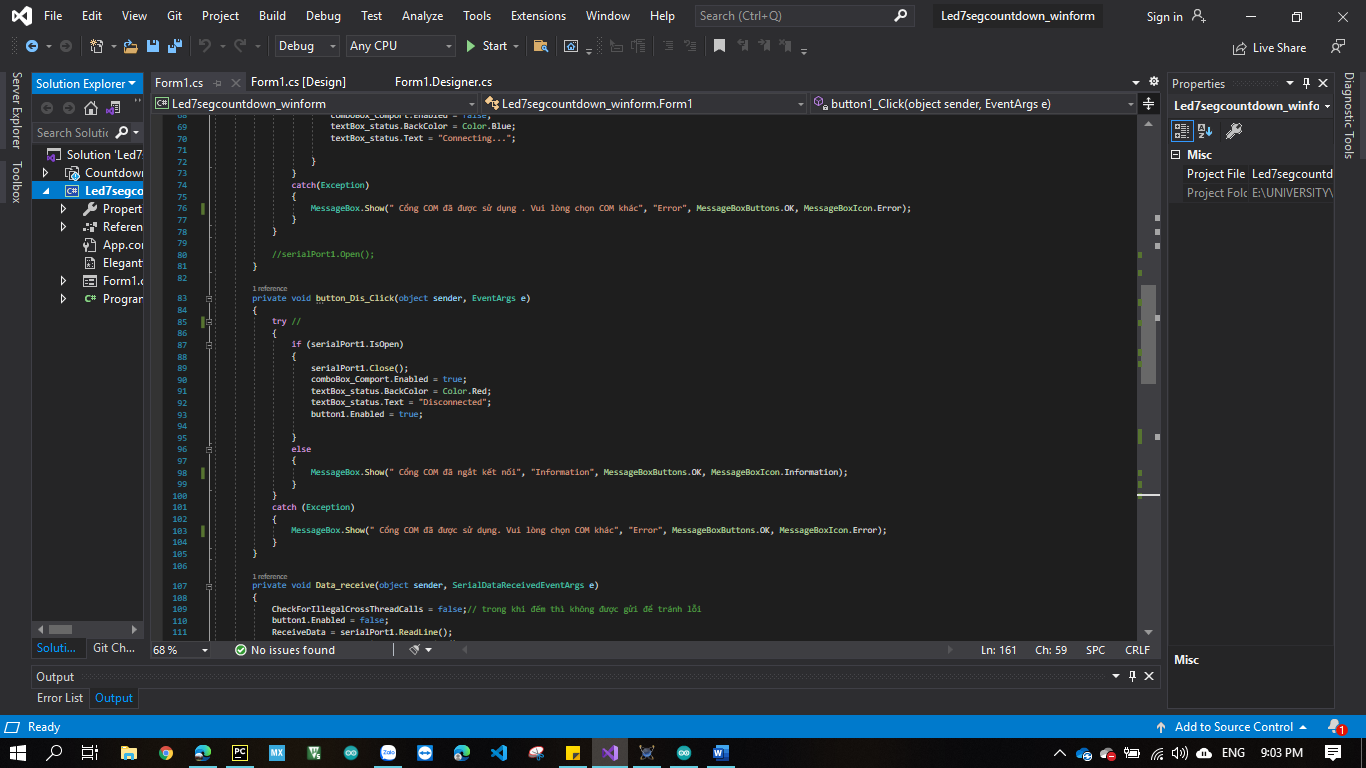
Một hình ảnh chụp màn hình Full – có code trong phần mềm MikroC.



**Hình 2.4: Lập trình trên mikcro C**

***2.4. Khả năng sử dụng phần mềm viết giao diện Visual Studio C#.***

Hai hình ảnh chụp màn hình Full – code và giao diện VS C#.



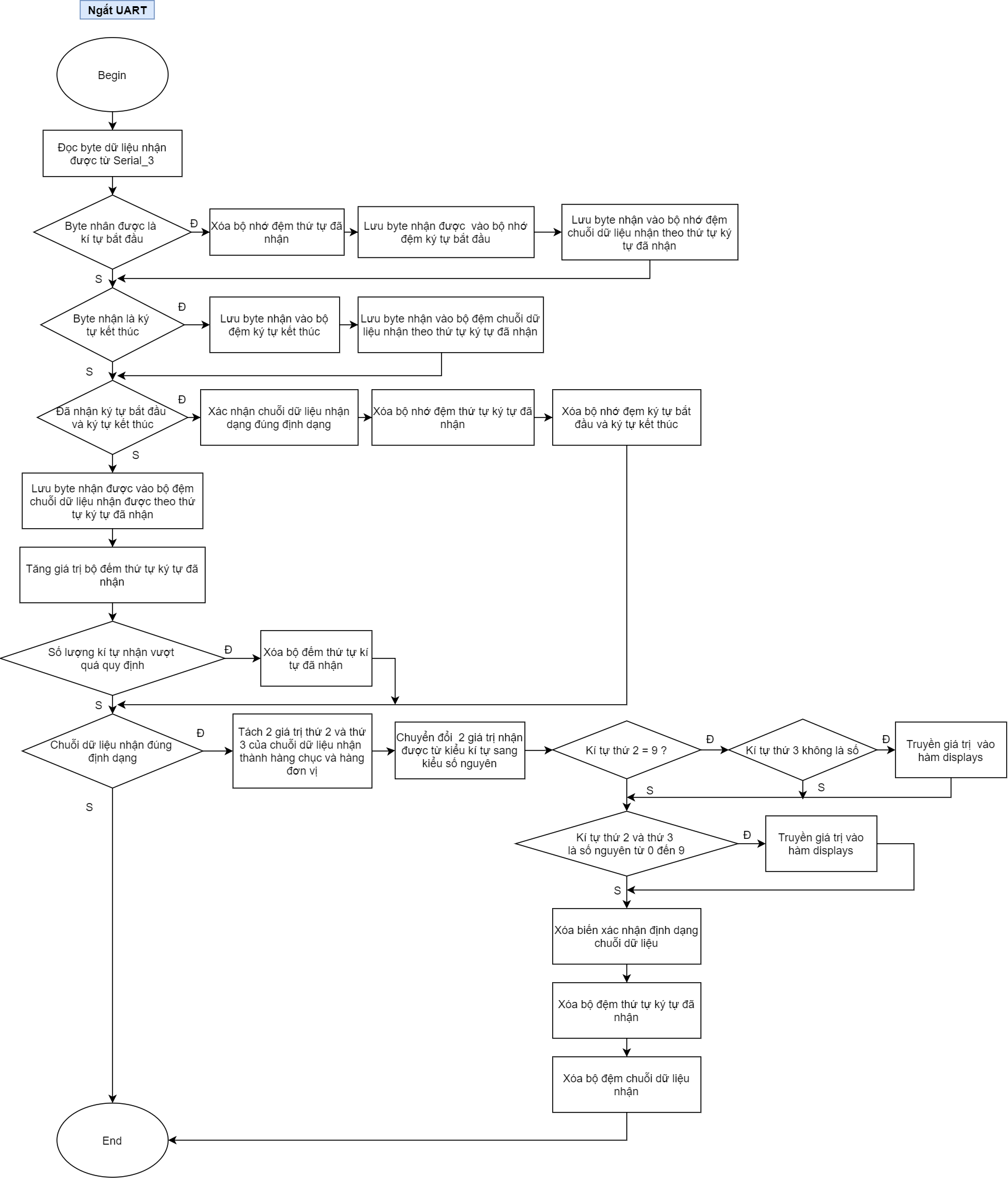
**Hình 2.5: Lập trình C# trên Visual Studio**

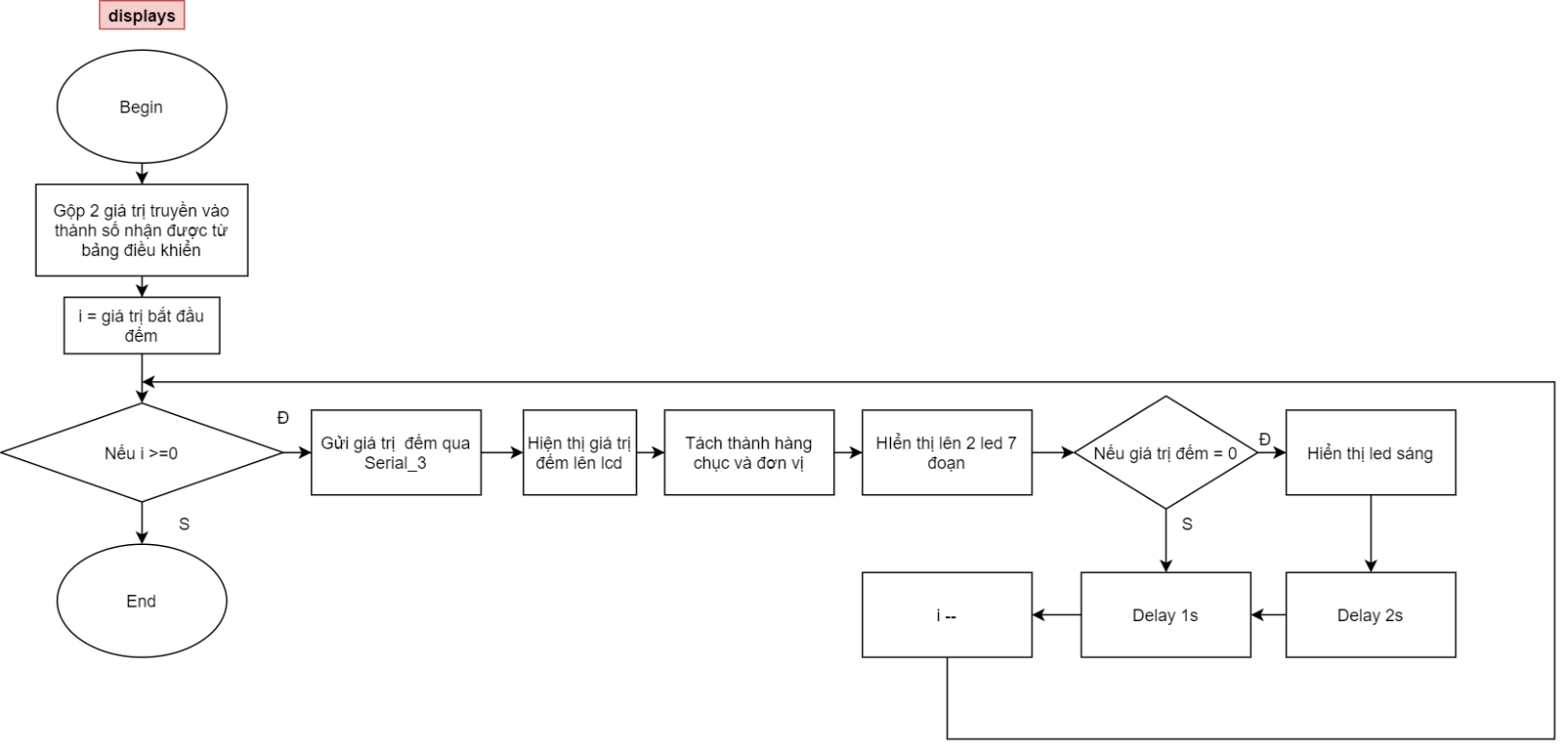
**PHẦN III. Chuẩn đầu ra môn học 3: Có khả năng phân tích, thiết kế giải thuật và lập trình ứng dụng**

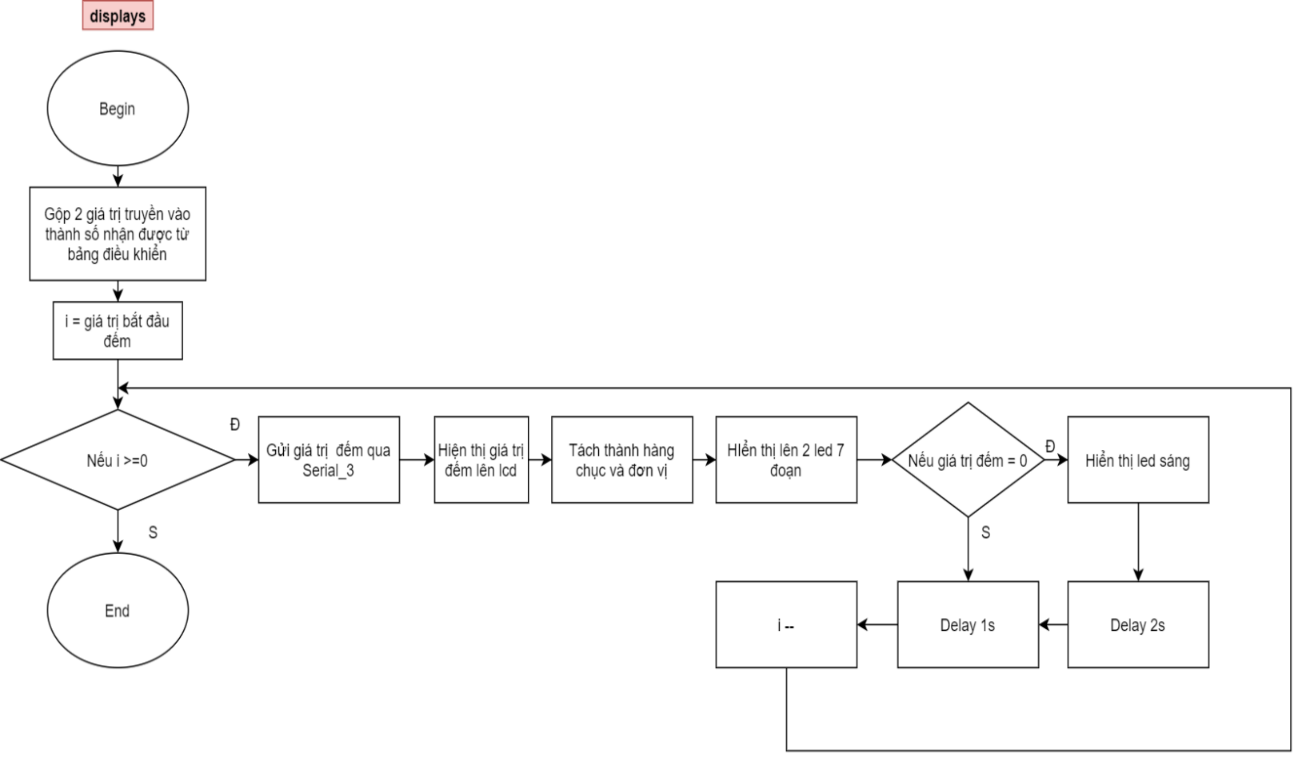
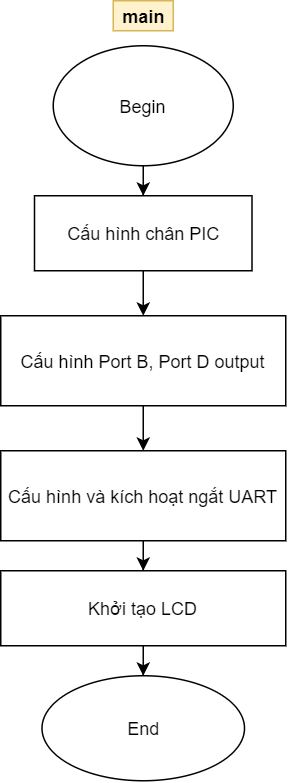
***3.1. Lưu đồ giải thuật cho Arduino hoặc PIC***

Lưu đồ giải thuật cho Arduino.







**Lưu đồ giải thuật cho PIC**



***3.2. Lập trình code cho Arduino hoặc PIC***

**Code adruino :**

#include <LiquidCrystal.h>

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

LiquidCrystal lcd(35, 39, 27, 29, 31, 33);

#define max\_count\_ReceiveData 4

char ReceiveData;

char start\_sign='@';

char end\_sign ='&';

char start\_Data, end\_Data;

int receive\_complete=0;

int count\_ReceiveData = 0;

char buf\_string\_receive[max\_count\_ReceiveData];

int datachar1;

int dataint1;

int datachar2;

int dataint2;

int n = 0;

int A = 7;

int B = 6;

int C = 4;

int D = 3;

int E = 2;

int F = 8;

int G = 9;

int Aa = 23;

int Ba = 25;

int Ca = 16;

int Da = 17;

int Ea = 18;

int Fa = 19;

int Ga = 20;

// mã hex của led 7 đoạn

byte num0 = 0xC0;

byte num1 = 0xF9;

byte num2 = 0xA4;

byte num3 = 0xB0;

byte num4 = 0x99;

byte num5 = 0x92;

byte num6 = 0x82;

byte num7 = 0xF8;

byte num8 = 0x80;

byte num9 = 0x90;

int num[] = {num0 ,num1, num2, num3, num4, num5, num6, num7, num8, num9};

void on(byte num)

{

int result = bitRead(num, 0);

if (result == 1)

{

digitalWrite(A, HIGH);}

else

{

digitalWrite(A, LOW);}

result = bitRead( num, 1);

if (result == 1)

{

digitalWrite(B, HIGH);}

else

{

digitalWrite(B, LOW);}

result = bitRead( num, 2);

if (result == 1)

{

digitalWrite(C, HIGH);}

else

{

digitalWrite(C, LOW);}

result = bitRead( num, 3);

if (result == 1)

{

digitalWrite(D, HIGH);}

else

{

digitalWrite(D, LOW);}

result = bitRead( num, 4);

if (result == 1)

{

digitalWrite(E, HIGH);}

else

{

digitalWrite(E, LOW);}

result = bitRead( num, 5);

if (result == 1)

{

digitalWrite(F, HIGH);}

else

{

digitalWrite(F, LOW);}

result = bitRead( num, 6);

if (result == 1)

{

digitalWrite(G, HIGH);}

else

{

digitalWrite(G, LOW);}

}

void on2(byte num)

{

int result = bitRead(num, 0);

if (result == 1)

{digitalWrite(Aa, HIGH);}

else

{digitalWrite(Aa, LOW);}

result = bitRead( num, 1);

if (result == 1)

{digitalWrite(Ba, HIGH);}

else

{digitalWrite(Ba, LOW);}

result = bitRead( num, 2);

if (result == 1)

{digitalWrite(Ca, HIGH);}

else

{digitalWrite(Ca, LOW);}

result = bitRead( num, 3);

if (result == 1)

{digitalWrite(Da, HIGH);}

else

{digitalWrite(Da, LOW);}

result = bitRead( num, 4);

if (result == 1)

{digitalWrite(Ea, HIGH);}

else

{digitalWrite(Ea, LOW);}

result = bitRead( num, 5);

if (result == 1)

{digitalWrite(Fa, HIGH);}

else

{digitalWrite(Fa, LOW);}

result = bitRead( num, 6);

if (result == 1)

{digitalWrite(Ga, HIGH);}

else

{digitalWrite(Ga, LOW);}

}

void setup() {

pinMode(A, OUTPUT);

pinMode(B, OUTPUT);

pinMode(C, OUTPUT);

pinMode(D, OUTPUT);

pinMode(E, OUTPUT);

pinMode(F, OUTPUT);

pinMode(G, OUTPUT);

pinMode(10,OUTPUT);

pinMode(11,OUTPUT);

pinMode(Aa, OUTPUT);

pinMode(Ba, OUTPUT);

pinMode(Ca, OUTPUT);

pinMode(Da, OUTPUT);

pinMode(Ea, OUTPUT);

pinMode(Fa, OUTPUT);

pinMode(Ga, OUTPUT);

Serial3.begin(9600,SERIAL\_8N1);

Serial.begin(9600);

lcd.begin(16,2);

}

void displays(int n,int x)

{

int countfrom = n \* 10 + x;

for(int i = countfrom; i>=0; i--)// thay i = uint8\_t

{

Serial3.println(i);

lcd.setCursor(7,0);

lcd.print(i);

int hc = i/10;

int dv = i%10;

on(num[hc]);

on2(num[dv]);

if( i == 0)

{

digitalWrite(11,HIGH);

delay(2000);

}

delay(1000);

digitalWrite(11,LOW);

}

n = 0;

}

void \_send()

{

if( digitalRead(11) == 0)

{

n = Serial3.read();

Serial3.print("n = ");

Serial3.println(int(n));

}

}

void serialEvent3()

{

// Doc du lieu moi lan nhan

ReceiveData = Serial3.read(); // Du kieu duoc truyen vao theo tung byte moi lan uart\_interrupt xay ra

// Kiem tra ki tu bat dau

if(ReceiveData == start\_sign)

{

count\_ReceiveData = 0;

start\_Data = ReceiveData;

buf\_string\_receive[count\_ReceiveData]= ReceiveData;

}

// Kiem tra ki tu ket thuc

if(ReceiveData == end\_sign)

{

end\_Data = ReceiveData;

buf\_string\_receive[count\_ReceiveData]= ReceiveData;

}

// Kiem tra done giu lieu da day du dinh dang ket thuc va bat dau

if((start\_Data == start\_sign)&&( end\_Data == end\_sign))

{

receive\_complete=1;

count\_ReceiveData=0;

start\_Data=NULL;

end\_Data=NULL;

}

else

{

// Neu du lieu khong dung ki tu bat dau

if(buf\_string\_receive[0] != start\_sign)

{

Clear\_buf\_string\_receive();

count\_ReceiveData = 0;

}

buf\_string\_receive[count\_ReceiveData] = ReceiveData;

count\_ReceiveData++;

if(count\_ReceiveData == max\_count\_ReceiveData)// neu count > max --> xoa count = 0 but phan du tiep tuc cong tiep tu 0 + len

{

count\_ReceiveData = 0;

receive\_complete = 0;

Clear\_buf\_string\_receive();

}

}

if(receive\_complete == 1)

{

// chuyen doi kieu char thanh int

datachar1 = buf\_string\_receive[1];

dataint1 = datachar1 - 48;

datachar2 = buf\_string\_receive[2];

dataint2 = datachar2 - 48;

if(dataint1 >= 9)

{

if(dataint2 > 9 || dataint2 < 0)

{

//Serial3.println(dataint1);

displays(0,dataint1);

}

}

if(dataint1 >= 0 && dataint1 <= 9 && dataint2 >=0 && dataint2 <=9)

{

//Serial3.println(dataint1);

// Serial3.println(dataint2);

displays(dataint1,dataint2);

}

receive\_complete = 0;

count\_ReceiveData = 0;

Clear\_buf\_string\_receive();

}

}

void Clear\_buf\_string\_receive()

{

for(int i=0; i<=max\_count\_ReceiveData; i++)

{

buf\_string\_receive[i]=NULL;

}

}

void loop() { // Loops forever

lcd.setCursor(1,0);

lcd.print("Count: ");

}

**Code PIC:**

#define led\_on 1

#define led\_off 0

#define max\_count\_ReceiveData 10

#define len\_code 5

// LCD pin

sbit LCD\_RS at RA0\_bit;

sbit LCD\_EN at RA1\_bit;

sbit LCD\_D4 at RC0\_bit;

sbit LCD\_D5 at RC1\_bit;

sbit LCD\_D6 at RC2\_bit;

sbit LCD\_D7 at RD7\_bit;

sbit LCD\_RS\_Direction at TRISA0\_bit;

sbit LCD\_EN\_Direction at TRISA1\_bit;

sbit LCD\_D4\_Direction at TRISC0\_bit;

sbit LCD\_D5\_Direction at TRISC1\_bit;

sbit LCD\_D6\_Direction at TRISC2\_bit;

sbit LCD\_D7\_Direction at TRISD7\_bit;

char txt1[] = "count: ";

char ReceiveData;

char buffer[5];

char buffer1[2];

int count = 0;

char led\_on\_code[] = "le\_on";

char led\_off\_code[] = "le\_off";

char start\_sign = '@';

char end\_sign = '&';

char count\_ReceiveData = 0;

char buf\_string\_receive[max\_count\_ReceiveData];

char start\_Data, end\_Data;

char receive\_complete = 0;

// Tao bang ma led 7 doan

const unsigned char led7segg[] = {0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82, 0xF8, 0x80, 0x90};

int datachar1;

int dataint1;

int datachar2;

int dataint2;

void displays(int n,int x)

{

int countfrom = (n \* 10 + x);

int i;

for( i = countfrom; i>=0; i--)

{

int hc = i/10;

int dv = i%10;

sprintf(buffer,"%d&",i);

UART\_Write\_Text(buffer);

sprintf(buffer1,"%d ",i);

Lcd\_Out(1,1,txt1);

Lcd\_Out(1,8,buffer1);

PORTD = led7segg[i/10];

PORTB = led7segg[i%10];

if (i == 0)

{

PORTB.RB7 = 0;

delay\_ms(2000);

}

delay\_ms(1000);

PORTB.RB7 = 1;

}

}

//chuong trinh xoa noi dung

void Clear\_buf\_string\_receive(void)

{

unsigned char i;

//chuong trinh

for(i=0;i<max\_count\_ReceiveData;i++)

{

buf\_string\_receive[i] = '\0';

}

}

//chuong trinh ngat

void interrupt(void)

{

if(PIR1.RCIF == 1)

{

PIR1.RCIF = 0;

ReceiveData = UART\_Read();

if(ReceiveData == start\_sign)

{

count\_ReceiveData = 0;

start\_Data = ReceiveData;

buf\_string\_receive[count\_ReceiveData]= ReceiveData;

}

// Kiem tra ki tu ket thuc

if(ReceiveData == end\_sign)

{

end\_Data = ReceiveData;

buf\_string\_receive[count\_ReceiveData]= ReceiveData;

}

// Kiem tra done giu lieu da day du dinh dang ket thuc va bat dau

if((start\_Data == start\_sign)&&( end\_Data == end\_sign))

{

receive\_complete=1;

count\_ReceiveData=0;

start\_Data= '\0';

end\_Data= '\0';

}

else

{

// Neu du lieu khong dung ki tu bat dau

if(buf\_string\_receive[0] != start\_sign)

{

Clear\_buf\_string\_receive();

count\_ReceiveData = 0;

}

buf\_string\_receive[count\_ReceiveData] = ReceiveData;

count\_ReceiveData++;

if(count\_ReceiveData == max\_count\_ReceiveData)

{

count\_ReceiveData = 0;

receive\_complete = 0;

Clear\_buf\_string\_receive();

}

}

if(receive\_complete == 1)

{

// chuyen doi kieu du lieu char thanh int

datachar1 = buf\_string\_receive[1];

dataint1 = datachar1 - 48;

datachar2 = buf\_string\_receive[2];

dataint2 = datachar2 - 48;

if(dataint1 >= 9)

{

if(dataint2 > 9 || dataint2 < 0)

{

displays(0,dataint1);

}

}

if(dataint1 >= 0 && dataint1 <= 9 && dataint2 >=0 && dataint2 <=9)

{

displays(dataint1,dataint2);

}

receive\_complete = 0;

count\_ReceiveData = 0;

Clear\_buf\_string\_receive();

}

}

}

void main()

{

unsigned char i = 0;

ADCON1 |= 0x0F;

CMCON |= 7;

//cau hinh Port B

TRISD = 0x00;

PORTD = 0x00;

TRISB = 0x00;

PORTB = 0x80;

//cau hinh Port E

PORTE = 0x00; LATE = 0x00;

TRISE.TRISE1 = 0;

//cau hinh ngat

INTCON.INT0IF = 0;

INTCON.INT0IE = 1;

INTCON2.INTEDG0 = 1;

PIR1.RCIF = 0;

PIE1.RCIE = 1;

INTCON.GIE = 1;

INTCON.PEIE = 1;

UART1\_Init(9600);

delay\_ms(100);

Lcd\_Init();

while(1);

}

***3.3. Lập trình giao diện VS C#***

***Code giao diện điều khiển C# cho Arduino:***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

using System.IO.Ports;

using System.Threading;

using System.Drawing;

namespace Led7segcountdown\_winform

{

public partial class Form1 : Form

{

byte a;

byte hc, dv;

byte b;

string TransmitData = string.Empty;

string ReceiveData = string.Empty;

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

string[] ports = SerialPort.GetPortNames();

foreach(string port in ports)

{

comboBox\_Comport.Items.Add(port);

}

}

private void comboBox\_Comport\_SelectedIndexChanged(object sender, EventArgs e)

{

serialPort1.PortName = comboBox\_Comport.Text;

}

private void button\_Connect\_Click(object sender, EventArgs e)

{

if(comboBox\_Comport.Text == "")

{

MessageBox.Show(" Vui lòng chọn cổng COM", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

else

{

try

{

if(serialPort1.IsOpen)

{

MessageBox.Show(" Cổng COM đã được kết nối ", "Information", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

serialPort1.Open();

comboBox\_Comport.Enabled = false;

textBox\_status.BackColor = Color.Blue;

textBox\_status.Text = "Connecting...";

}

}

catch(Exception)

{

MessageBox.Show(" Cổng COM đã được sử dụng . Vui lòng chọn COM khác", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

private void button\_Dis\_Click(object sender, EventArgs e)

{

try //

{

if (serialPort1.IsOpen)

{

serialPort1.Close();

comboBox\_Comport.Enabled = true;

textBox\_status.BackColor = Color.Red;

textBox\_status.Text = "Disconnected";

button1.Enabled = true;

}

else

{

MessageBox.Show(" Cổng COM đã ngắt kết nối", "Information", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception)

{

MessageBox.Show(" Cổng COM đã được sử dụng. Vui lòng chọn COM khác", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void Data\_receive(object sender, SerialDataReceivedEventArgs e)

{

CheckForIllegalCrossThreadCalls = false;

button1.Enabled = false;

ReceiveData = serialPort1.ReadLine();

textBox3.Text = ReceiveData.ToString();

b = byte.Parse(textBox3.Text);

hc = (byte)(b / 10);

dv = (byte)(b % 10);

if (b == 0)

{

led7Segment1.LEDNumber = hc;

led7Segment2.LEDNumber = dv;

Led\_State.BackColor = Color.Red;

Thread.Sleep(3000);

Led\_State.BackColor = Color.LightYellow;

button1.Enabled = true;

}

led7Segment1.LEDNumber = hc;

led7Segment2.LEDNumber = dv;

}

private void button1\_Click(object sender, EventArgs e)

{

try //

{

if (serialPort1.IsOpen)

{

a = byte.Parse(textBox1.Text);

if (a <= 99 && a >= 9)

{

TransmitData = "@" + a.ToString() + "&";

serialPort1.Write(TransmitData);

textBox2.Text = TransmitData;

}

else

{

MessageBox.Show("Vui lòng nhập số trong khoảng 9 - 99", "Information", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

else

{

MessageBox.Show(" Cổng COM chưa được kết nối. Vui lòng kiểm tra lại", "Information", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception)

{

MessageBox.Show(" Cổng COM đã được sử dụng. Vui lòng chọn COM khác", "Information", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

***Code giao diện điều khiển C# cho PIC:***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

using System.IO.Ports;

using System.Threading;

using System.Drawing;

namespace Led7segcountdown\_winform

{

public partial class Form1 : Form

{

byte a;

byte hc, dv;

byte b;

string TransmitData = string.Empty;

string ReceiveData = string.Empty;

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

string[] ports = SerialPort.GetPortNames();

foreach(string port in ports)

{

comboBox\_Comport.Items.Add(port);

}

}

private void comboBox\_Comport\_SelectedIndexChanged(object sender, EventArgs e)

{

serialPort1.PortName = comboBox\_Comport.Text;

}

private void button\_Connect\_Click(object sender, EventArgs e)

{

if(comboBox\_Comport.Text == "")

{

MessageBox.Show(" Vui lòng chọn cổng COM", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

else

{

try

{

if(serialPort1.IsOpen)

{

MessageBox.Show(" Cổng COM đã được kết nối ", "Information", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

serialPort1.Open();

comboBox\_Comport.Enabled = false;

textBox\_status.BackColor = Color.Blue;

textBox\_status.Text = "Connecting...";

}

}

catch(Exception)

{

MessageBox.Show(" Cổng COM đã được sử dụng . Vui lòng chọn COM khác", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

//serialPort1.Open();

}

private void button\_Dis\_Click(object sender, EventArgs e)

{

try //

{

if (serialPort1.IsOpen)

{

serialPort1.Close();

comboBox\_Comport.Enabled = true;

textBox\_status.BackColor = Color.Red;

textBox\_status.Text = "Disconnected";

button1.Enabled = true;

}

else

{

MessageBox.Show(" Cổng COM đã ngắt kết nối", "Information", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception)

{

MessageBox.Show(" Cổng COM đã được sử dụng. Vui lòng chọn COM khác", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void Data\_receive(object sender, SerialDataReceivedEventArgs e)

{

CheckForIllegalCrossThreadCalls = false;//

button1.Enabled = false;

ReceiveData = serialPort1.ReadTo("&");

textBox3.Text = ReceiveData.ToString();

b = byte.Parse(textBox3.Text);

hc = (byte)(b / 10);

dv = (byte)(b % 10);

if (b == 0)

{

led7Segment1.LEDNumber = hc;

led7Segment2.LEDNumber = dv;

//

Led\_State.BackColor = Color.Red;

Thread.Sleep(3000);

Led\_State.BackColor = Color.LightYellow;

button1.Enabled = true;

}

led7Segment1.LEDNumber = hc;

led7Segment2.LEDNumber = dv;

}

private void button1\_Click(object sender, EventArgs e)

{

try //

{

if (serialPort1.IsOpen)

{

a = byte.Parse(textBox1.Text);

if (a <= 99 && a >= 9)

{

TransmitData = "@" + a.ToString() + "&";

serialPort1.Write(TransmitData);

textBox2.Text = TransmitData;

}

else

{

MessageBox.Show("Vui lòng nhập số trong khoảng 9 - 99", "Information", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

else

{

MessageBox.Show(" Cổng COM chưa được kết nối. Vui lòng kiểm tra lại", "Information", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception)

{

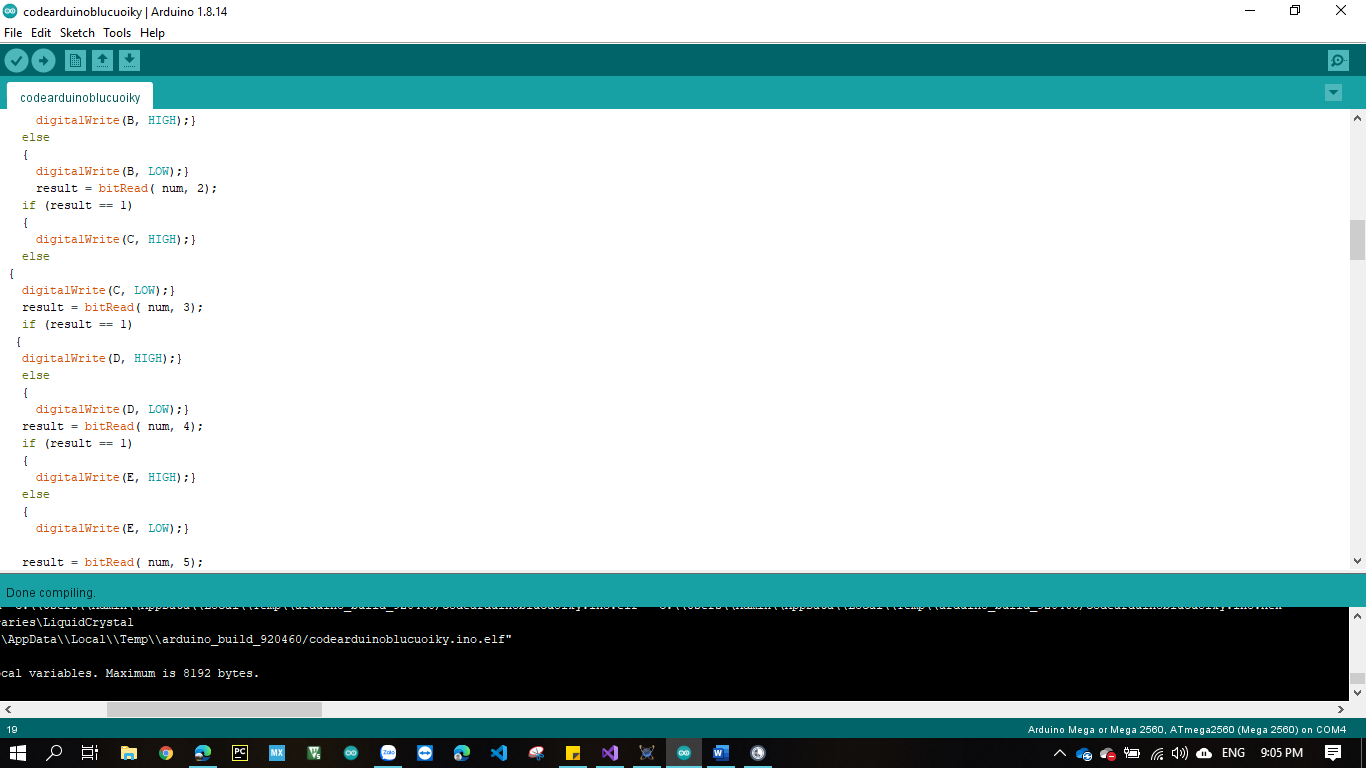
MessageBox.Show(" Cổng COM đã được sử dụng. Vui lòng chọn COM khác", "Information", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

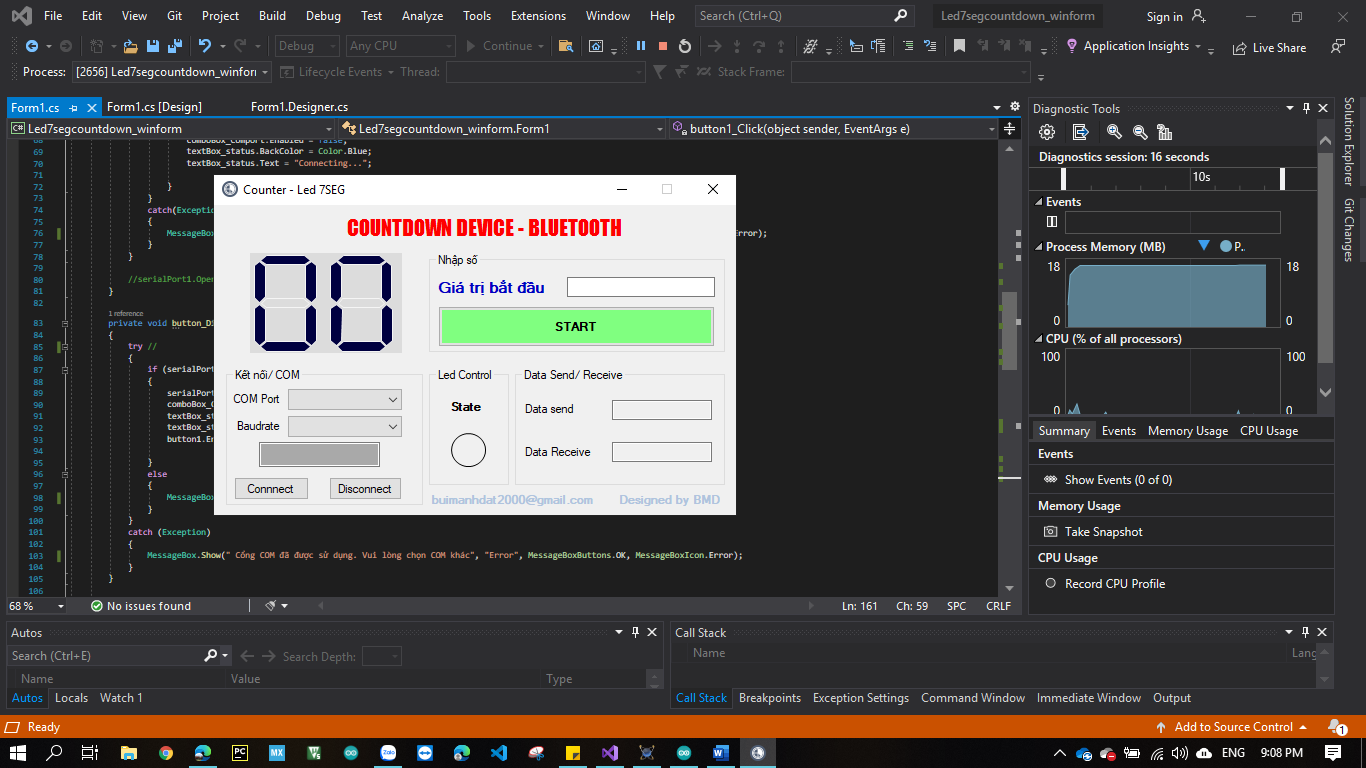
}

}

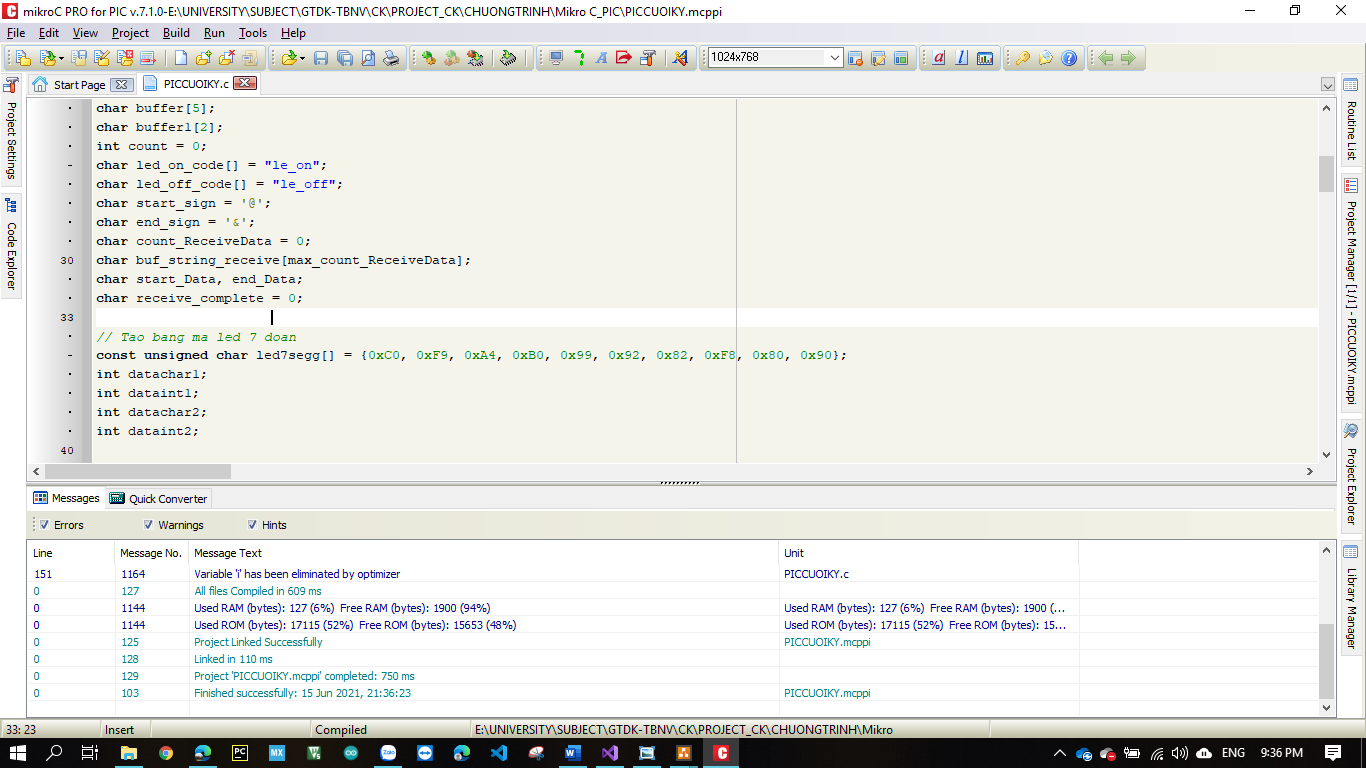
}

***3.4. Biên dịch code và sửa lỗi.***

**Hình 3.1: Chương trình Arduino**



**Hình 3.2: Chương trình giao diện C#**

**Hình 3.3: Chương trình cho PIC18F4550 trên Mikro C**

**PHẦN IV: Chuẩn đầu ra môn học 4: Có khả năng lắp ráp, cân chỉnh các board mạch điện tử theo qui trình**

***4.1. Quy trình Mô phỏng và Test thực nghiệm***

Các lệnh cấu hình module HC-05 mở chế độ slave

Vào chế độ command:

* AT : Kiểm tra đáp ứng
* AT+ORGL: Khôi phục về trạng thái ban đầu
* AT+NAME = HC05\_slave: Đặt tên cho mô-đun
* AT+UART=115200,0,0: Thiết lập thông số truyền thông
* AT+RMAAD: Ngắt kết nối với các thiết bị đã bắt cặp
* AT+PSWD=1234: Đặt lại mật khẩu
* AT+ROLE=0: Đặt mô-đun ở vai trò Slave
* AT+RESET: Khởi động lại mô-đun
* Đáp ứng: OK

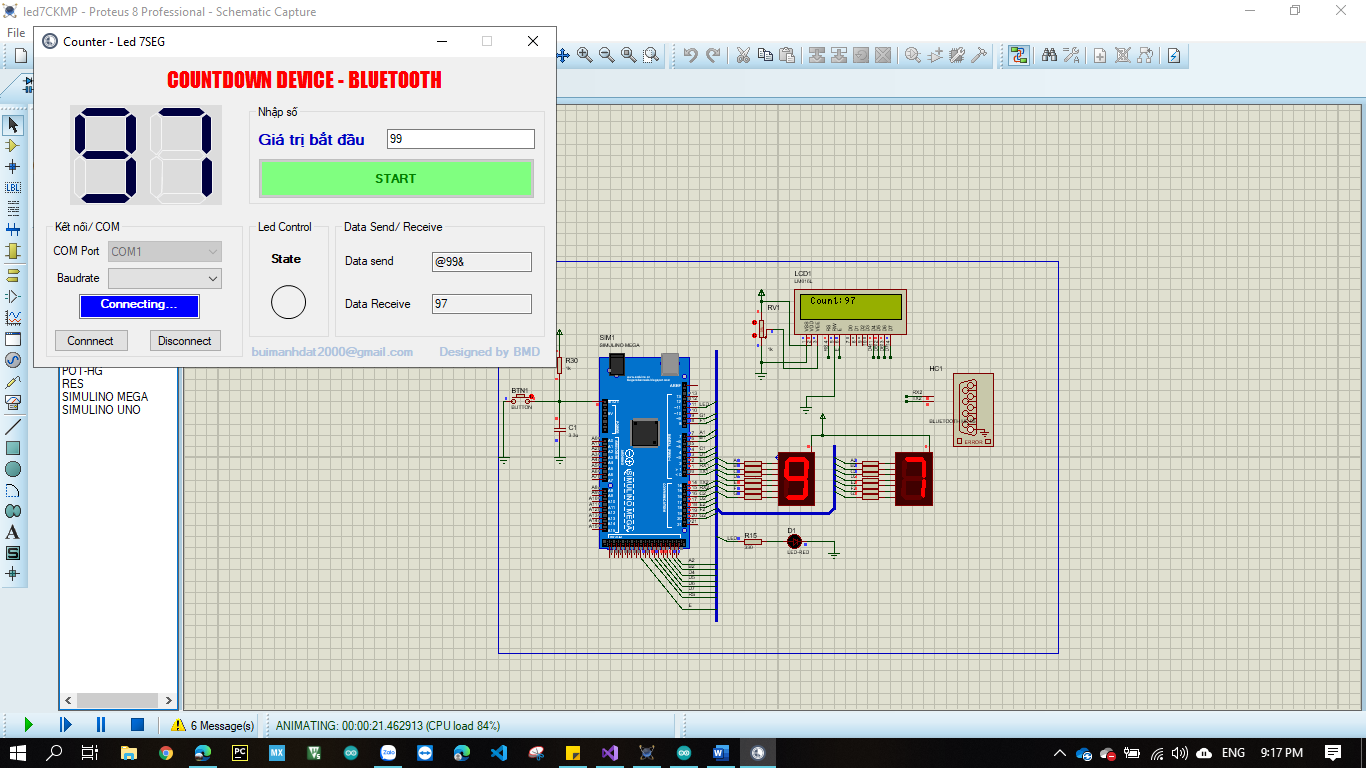
Biên dịch chương trình:

* Biên dịch chương trình và lấy file hex
* Chọn đường dẫn nạp file hex vào vi điều khiển trên proteus
* Chọn cổng COM 2 và tốc độ baudrate 9600
* Sử dụng phần mềm Virtual Serial Port Driver để tạo kết nối 2 COM ảo với nhau

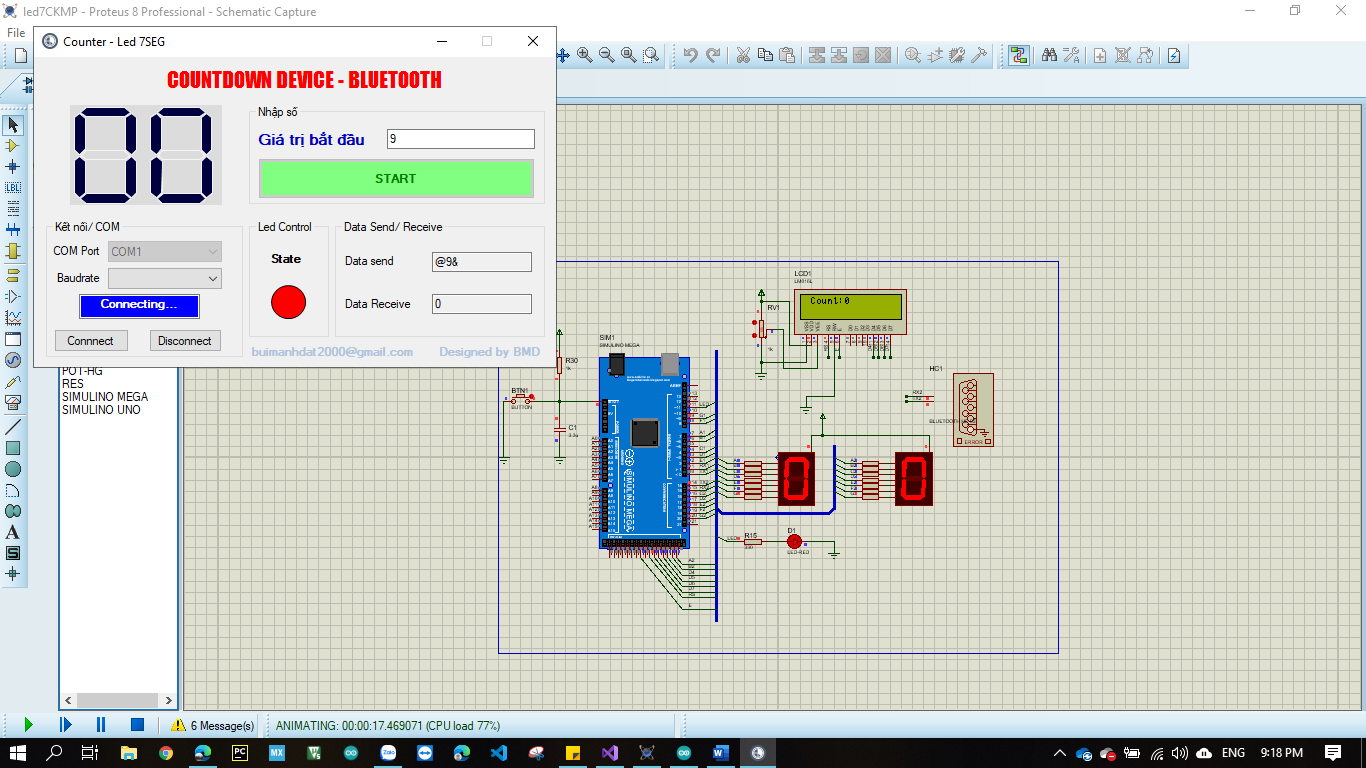
Tiến hành mô phỏng:

* Chọn cổng COM: COM1
* Nhập giá trị bắt đầu
* Nhấn nút START trên bảng điều khiển:
* Bảng điều khiển gửi giá trị đếm xuống
* Hiển thị giá trị đếm được lên LED ma trận
* Hiển thị dữ liệu gửi đi trên textbox Data send
* Hiện thị dữ liệu trả về (nhận được) lên textbox Data receive
* Khi giá trị nhận được về 00 led status sáng 3s rồi tắt
* Hiển thị giá trị đếm ra led 7 đoạn trên mô phỏng
* Khi giá trị đếm về 00 hiện thị led thông báo trạng thái sáng 3s

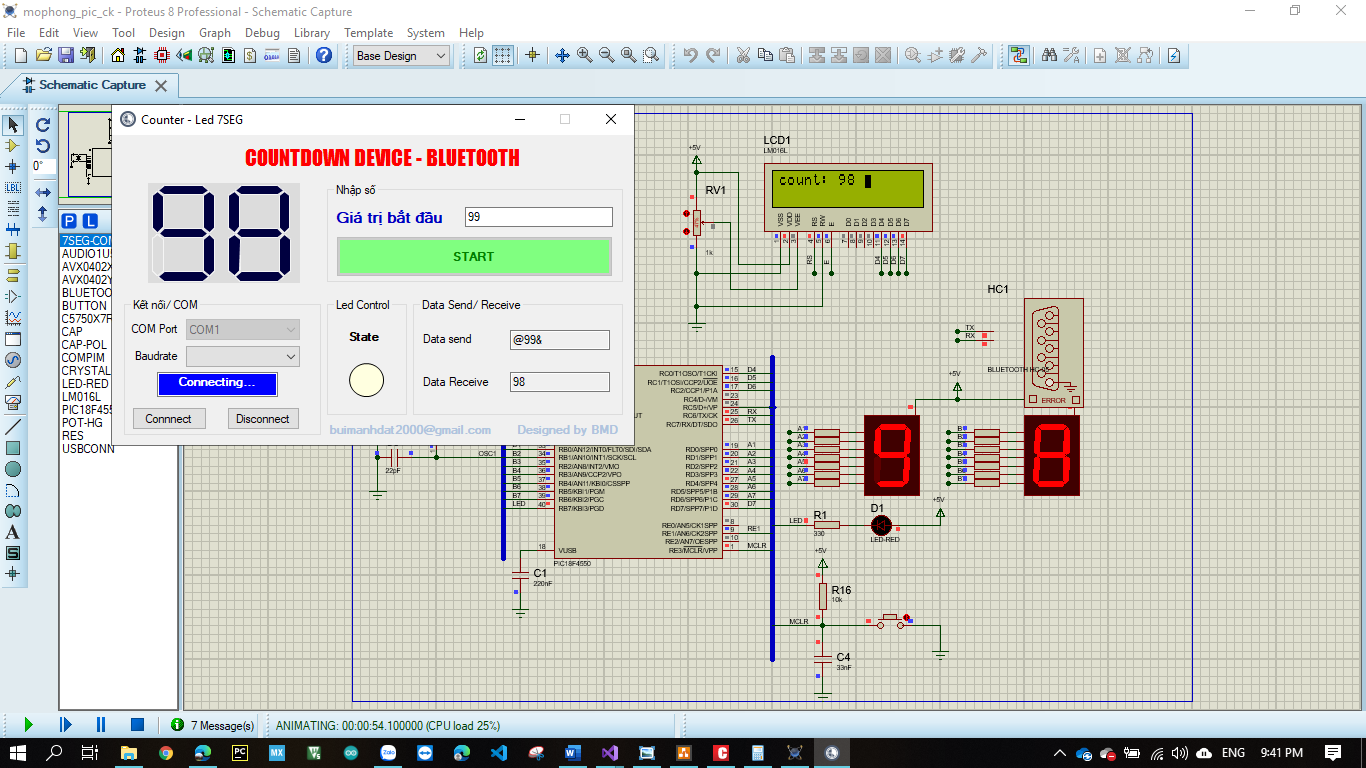
***4.2. Kết quả mô phỏng***



**Hình 4.1: Kết quả mô phỏng với Arduino mega**

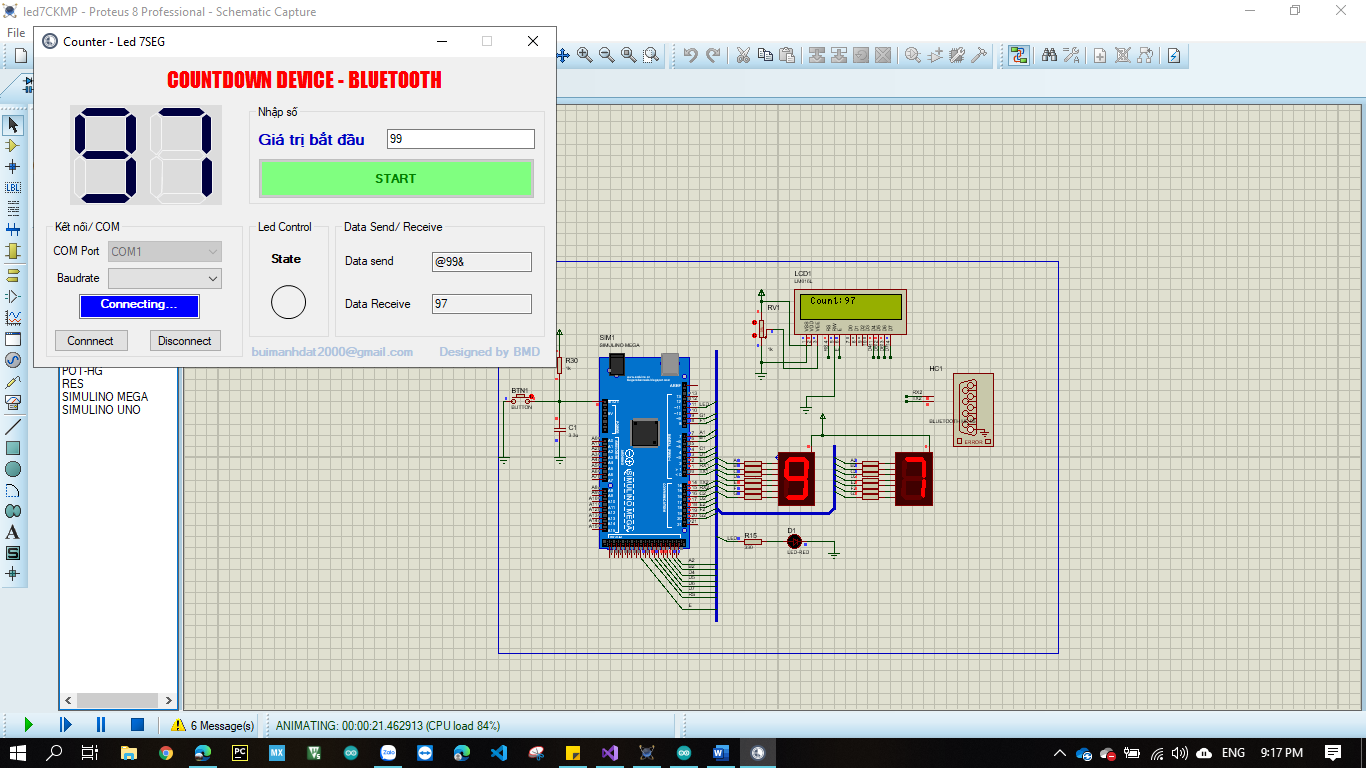


**Hình 4.2: Kết quả mô phỏng với Arduino mega**



**Hình 4.3 : Kết quả mô phỏng với PIC18F4550**

**PHẦN V: Kết luận bài thực hành**



* Kết quả mô phỏng đúng theo yêu cầu của đề bài