**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA ĐIỆN - ĐIỆN TỬ**

****

**BÀI TẬP LỚN MÔN HỆ THỐNG ĐIỀU KHIỂN NHÚNG**

**PWM VÀ ENCODER SỬ DỤNG STM32F407VG**

**Nhóm 9**

**GVHD: Nguyễn Vĩnh Hảo**

|  |  |
| --- | --- |
| **Sinh viên** | **MSSV** |
| Nguyễn Đình Thông | 1413833 |
| Nguyễn Văn An | 1410033 |
| Nguyễn Văn Bình | 1410293 |

**I. Kết nối và lập trình vi điều khiển STM32F4**

**1. PWM.**

**2. Encoder.**

**3. Ứng dụng điều khiển PID động cơ.**

**4. Lập trình PID.**

**5. UART.**

**II. Phần giao diện. Lập trình C# Win Form**

**A.Giao diện kết nối cổng COM**

**B. Giao diện điều khiển động cơ.**

**1. Tín hiệu điều khiển:**

a. Điều khiển vận tốc quay chiều thuận.

b. Điều khiển vận tốc quay chiều thuận.

c. Điều khiển vị trí.

**2. Hàm xử lý 1 số double thành string.**

**3. Hàm draw(double ypoint). Vẽ một điểm trên đồ thị.**

**4. Hàm nhận dữ liệu COM từ STM32 lên PC.**

**5. Hàm xử lý và gọi lệnh vẽ đồ thị.**

**Đề tài PWM và Encoder**

**Ứng dụng: điều khiển PID động cơ DC**

**I. Kết nối và lập trình vi điều khiển STM32F4**

**1. PWM.**

Cấu hình: trên chân PD15

Tần số cấp cho Timer là 1Mhz

Chu kì điều rộng xung là 20ms

Dùng Timer chế độ đếm lên 20000 nhịp

**/\* TIM4 init function \*/**

**void MX\_TIM4\_Init(void)**

**{**

**TIM\_MasterConfigTypeDef sMasterConfig;**

**TIM\_OC\_InitTypeDef sConfigOC;**

**htim4.Instance = TIM4;**

**htim4.Init.Prescaler = 84;**

**htim4.Init.CounterMode = TIM\_COUNTERMODE\_UP;**

**htim4.Init.Period = 20000;**

**htim4.Init.ClockDivision = TIM\_CLOCKDIVISION\_DIV1;**

**HAL\_TIM\_PWM\_Init(&htim4);**

**}**

**2. Encoder.**

Để đọc encoder, ta sử dụng ngắt ngoài khi có cạnh lên trên PA3 (xung A), input trên PA4 (xung B)

Cấu hình:

**void MX\_GPIO\_Init(void)**

**{**

**GPIO\_InitTypeDef GPIO\_InitStruct;**

**/\* GPIO Ports Clock Enable \*/**

**\_\_HAL\_RCC\_GPIOH\_CLK\_ENABLE();**

**\_\_HAL\_RCC\_GPIOA\_CLK\_ENABLE();**

**\_\_HAL\_RCC\_GPIOB\_CLK\_ENABLE();**

**\_\_HAL\_RCC\_GPIOD\_CLK\_ENABLE();**

**/\*Configure GPIO pin : PA3 \*/**

**GPIO\_InitStruct.Pin = GPIO\_PIN\_3;**

**GPIO\_InitStruct.Mode = GPIO\_MODE\_IT\_RISING;**

**GPIO\_InitStruct.Pull = GPIO\_NOPULL;**

**HAL\_GPIO\_Init(GPIOA, &GPIO\_InitStruct);**

**/\*Configure GPIO pin : PA4 \*/**

**GPIO\_InitStruct.Pin = GPIO\_PIN\_4;**

**GPIO\_InitStruct.Mode = GPIO\_MODE\_INPUT;**

**GPIO\_InitStruct.Pull = GPIO\_PULLUP;**

**HAL\_GPIO\_Init(GPIOA, &GPIO\_InitStruct);**

**/\* EXTI interrupt init\*/**

**HAL\_NVIC\_SetPriority(EXTI3\_IRQn, 0, 0);**

**HAL\_NVIC\_EnableIRQ(EXTI3\_IRQn);**

**}**

**3. Ứng dụng điều khiển PID động cơ.**

1. **Động cơ:** DC Servo 12V, tỉ số truyền: 1:9, tốc độ tối đa: 400 rpm.
2. **Encoder:** 334 xung.
3. **Cầu H:** L298D.
4. **Điều khiển vị trí:** thời gian lấy mẫu: 15ms.

Vị trí động cơ = số xung encoder x 360 / (334 x 9)

1. **Điều khiển tốc độ:** thời gian lấy mẫu: 15ms

Tốc độ động cơ = (số xung encoder(k) – số xung encoder(k-1)) / 15ms

**4. Lập trình PID.**

float KP , KI , KD , Ts=0.015, ek2 ,ek1 ,ek ,uk ,uk1 ,umax = 20000,umin = -20000 ,u ,setpoint, measure;

HAL\_Delay(15);

ek2 = ek1;

ek1 = ek;

ek = setpoint - measure;

uk1 = uk;

uk = uk1+KP\*(ek-ek1)+KI\*Ts\*(ek+ek1)/2+KD\*(ek-2\*ek1+ek2)/Ts;

if(uk > umax) uk = umax;

if(uk < umin) uk = umin;

if(uk < 0)

{

HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_13,1);

\_\_HAL\_TIM\_SET\_COMPARE(&htim4,TIM\_CHANNEL\_4,20000+uk);

}

else

{

HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_13,0);

\_\_HAL\_TIM\_SET\_COMPARE(&htim4,TIM\_CHANNEL\_4,uk);

}

**5. UART.**

Sử dụng USART3, không dùng DMA.

Tốc độ baud: 9600, data: 8 bit, stop bit: 1

Cấu hình:

void MX\_USART3\_UART\_Init(void)

{

huart3.Instance = USART3;

huart3.Init.BaudRate = 9600;

huart3.Init.WordLength = UART\_WORDLENGTH\_8B;

huart3.Init.StopBits = UART\_STOPBITS\_1;

huart3.Init.Parity = UART\_PARITY\_NONE;

huart3.Init.Mode = UART\_MODE\_TX\_RX;

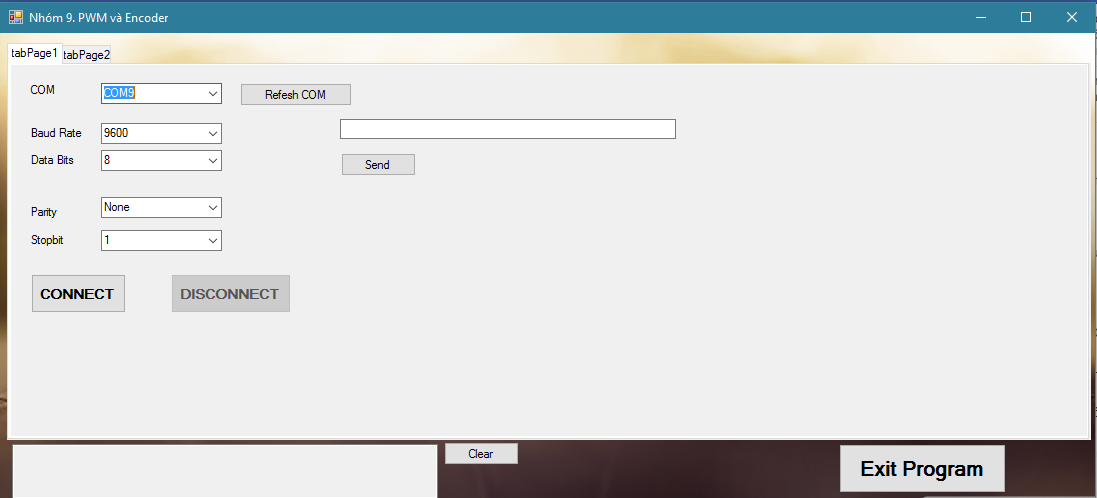
huart3.Init.HwFlowCtl = UART\_HWCONTROL\_NONE;

huart3.Init.OverSampling = UART\_OVERSAMPLING\_16;

HAL\_UART\_Init(&huart3);

}

**II. Phần giao diện. Lập trình C# Win Form**

**A. Giao diện kết nối cổng COM**

* Code thiết lập cổng COM:

      SerialPort P = new System.IO.Ports.SerialPort();

      //Declare a buffer for storing data

      string InputData = String.Empty;

public MotorControl()

   {

           InitializeComponent();

           string[] ports = SerialPort.GetPortNames();

           cbCom.Items.AddRange(ports);

           P.ReadTimeout = 1000;

           P.DataReceived += new SerialDataReceivedEventHandler(DataReceive);

           string[] BaudRate =

           {

               "1200", "2400", "4800", "9600", "19200",

               "38400", "57600", "115200"

           };

           cbRate.Items.AddRange(BaudRate);

           string[] Databits = {"6", "7", "8"};

           cbBits.Items.AddRange(Databits);

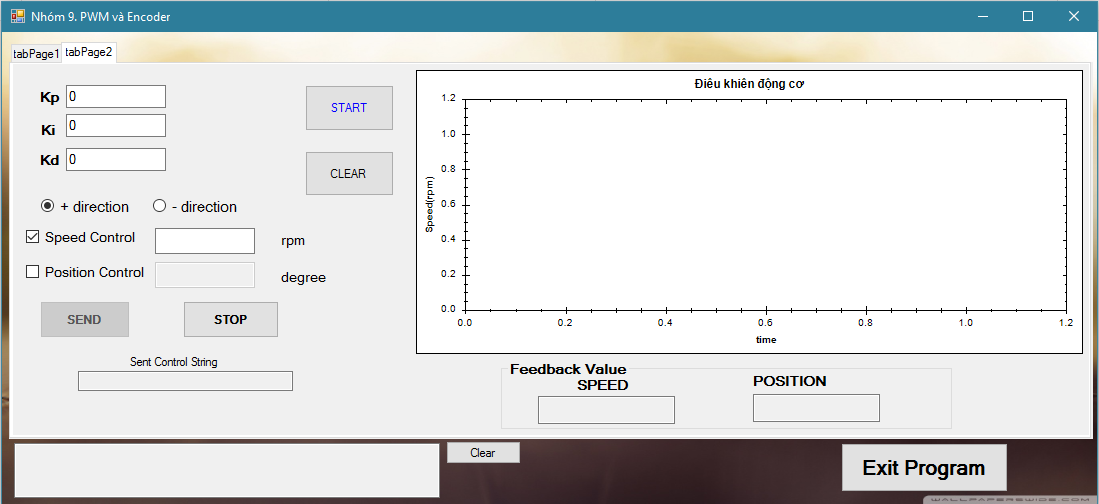
           string[] Parity = {"None", "Odd", "Even"};

           cbParity.Items.AddRange(Parity);

           string[] stopbit = {"1", "1.5", "2"};

           cbBit.Items.AddRange(stopbit);

       }

**B. Giao diện điều khiển động cơ.**

**1. Tín hiệu điều khiển:**

**a. Điều khiển vận tốc quay chiều thuận.**

@Pxxx.xxIxxx.xxDxxx.xxVxxx.xx

**b. Điều khiển vận tốc quay chiều thuận.**

!Pxxx.xxIxxx.xxDxxx.xxVxxx.xx

public void sendspeedcontrol(bool pdir,double kP,double kI,double kD,double sPEED,out string truyendi)

      {

          truyendi = "";

          if (pdir)

          {

              //send speed control with + direction to COM

              //send[0] = 40; '@' character

              truyendi += "@P";

          }

          else

          {

              //send speed control with - direction to COM

              //send[0] = 21; //'!' character

              truyendi += "!P";

          }

          truyendi+=numberprocessor(kP);

          truyendi += "I";

          truyendi += numberprocessor(kI);

          truyendi += "D";

          truyendi += numberprocessor(kD);

          truyendi += "V";

          truyendi += numberprocessor(sPEED);

      }

**c. Điều khiển vị trí.**

@Pxxx.xxIxxx.xxDxxx.xxXxxx.xx

public void sendpositioncontrol(double kP, double kI, double kD, double pOSITION, out string truyendi)

{

    truyendi = "@P";

    truyendi += numberprocessor(kP);

    truyendi += "I";

    truyendi += numberprocessor(kI);

    truyendi += "D";

    truyendi += numberprocessor(kD);

    truyendi += "X";

    truyendi += numberprocessor(pOSITION);

}

**2. Hàm xử lý 1 số double thành string.**

public string numberprocessor(double num)

       {

           if (num == 0)

               return "000.00";

           string vip="",returnstring="";

           string xtring;

           if (num >= 100 && num < 1000)

           {

               vip+= Convert.ToString(num);

           }

           else if (num>=10 && num<100)

           {

               vip+="0"+ Convert.ToString(num);

           }

           else

           {

               vip+= "00" + Convert.ToString(num);

           }

           returnstring += vip.Substring(0, 3) + ".";

           xtring = Convert.ToString(num \* 100);

           returnstring += xtring.Substring(xtring.Length - 2, 2);

           return returnstring;

       }

**3. Hàm draw(double ypoint). Vẽ một điểm trên đồ thị.**

public void draw(double ypoint)

     {

         zedGraphControl1.GraphPane.CurveList.Clear();

         // Time được tính bằng ms

         double time = (Environment.TickCount - tickStart) / 1000.0;

         //add point to Zed Graph

         listPointsOne.Add(time, ypoint);

         myCurveOne = myPane.AddCurve(null, listPointsOne, Color.Red, SymbolType.Default);

         zedGraphControl1.AxisChange();

         zedGraphControl1.Invalidate();

     }

Ngay khi nhận dữ liệu từ cổng COM từ STM32 lên PC. Ta kiểm tra xem cú pháp của dữ liệu có hợp lệ hay không. Nếu hợp lệ thì sẽ hiển thi lên ô SPEED đồng thời vẽ lên đồ thị.

**4. Hàm nhận dữ liệu COM từ STM32 lên PC.**

private void SetText(string text)

       {

           COMhandler(text);

           if (this.txtkq.InvokeRequired)

           {

               SetTextCallback d = new SetTextCallback(SetText);

               this.Invoke(d, new object[] { text });

           }

           else

           {

               if (text == "$")

               {

                   COMhandler(txtkq.Text);

                   txtkq.Text = "";

               }

               else

               {

                   txtkq.Text += text;

               }

           }

       }

**5. Hàm xử lý và gọi lệnh vẽ đồ thị.**

public void COMhandler(string data)

       {

           try

           {

               double fbspeed, fbposition;

               bool checkfbspeed, checkfbposition;

               checkfbspeed = Double.TryParse(data.Substring(0, 7), out fbspeed);

               checkfbposition = Double.TryParse(data.Substring(7, 5), out fbposition);

               if (checkfbspeed)

               {

                   txtsp.Text = Convert.ToString(fbspeed);

               }

               if (checkfbposition)

               {

                   txtpo.Text = Convert.ToString(fbposition);

               }

               if (checkspeed.CheckState == CheckState.Checked)

               {

                   myPane.YAxis.Title.Text = "Speed(rpm)";

                   draw(fbspeed);

               }

               else

               {

                   myPane.YAxis.Title.Text = "Vi tri ( degree )";

                   draw(fbposition);

               }

           }

           catch (Exception e)

           {

               Console.WriteLine(e);

           }

       }