**TEST THƯ VIỆN**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Threading;

using PLCPiProject;

namespace PLCPiTestConso

{

class Program

{

static void Main()

{

PLCPi myPLCPi = new PLCPi();

while (true)

{

Console.WriteLine("0\_on");

myPLCPi.Output("R0", 0xf0);

Thread.Sleep(200);

Console.WriteLine("1\_on");

myPLCPi.Output("R1", 0x0f);

Thread.Sleep(200);

Console.WriteLine("0\_off");

myPLCPi.Output("R0", 0);

Thread.Sleep(200);

Console.WriteLine("1\_off");

myPLCPi.Output("R1", 0);

Thread.Sleep(200);

/\*Console.WriteLine("0\_on");

myPLCPi.Output("R1.0", 1);

Thread.Sleep(200);

Console.WriteLine("1\_on");

myPLCPi.Output("R1.1", 1);

Thread.Sleep(200);

Console.WriteLine("0\_on");

myPLCPi.Output("R1.2", 1);

Thread.Sleep(200);

Console.WriteLine("1\_on");

myPLCPi.Output("R1.3", 1);

Thread.Sleep(200);

Console.WriteLine("0\_on");

myPLCPi.Output("R1.4", 1);

Thread.Sleep(200);

Console.WriteLine("1\_on");

myPLCPi.Output("R1.5", 1);

Thread.Sleep(200);

Console.WriteLine("0\_on");

myPLCPi.Output("R1.6", 1);

Thread.Sleep(200);

Console.WriteLine("1\_on");

myPLCPi.Output("R1.7", 1);

Thread.Sleep(200);\*/

}

}

}

}

**PLCPi**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Threading;

namespace PLCPiProject

{

public class PLCPi

{

Out\_Class myOut\_Class = new Out\_Class();

//Method Input

//3 dấu xẹt là để viết ghi chú, để khi người ta dùng đối tượng của lớp này, vừa gọi hàm này

//thì nó sẽ xuất hiện nội dung gợi ý ra chính là các ghi chú ở đây, người ta đọc sẽ hiểu là đang làm gì

//PLCPi thì em ghi chú băng tiếng Việt luôn

/// <summary>

/// Phương thức này dùng để đọc giá trị của 1 dạng ngõ vào về

/// </summary>

/// <param name="Type">Tên ngõ vào cần đọc. "V0.0": đọc bit, "V0": đọc byte</param>

/// <returns></returns>

public string Input(string Type)

{

//Code here

return null;//trả rỗng về, em lập trình trong này rồi trả về sau

}

/// <summary>

/// xuất ngõ ra. có 2 kiểu xuất. xuất theo Bit và theo Byte

/// </summary>

/// <param name="Type">Type = "Rx" thì xuấy Byte. Type = "Rx.x" xuất Bit</param>

/// <param name="Value">xuất Byte thì Value = 1Byte. xuất Bit thì Value = 0 or 1</param>

public void Output(string Type, int Value)

{

if (Type.Length == 4)//xuất bit'r0.0'

{

if (Type.Substring(0, 2) == "R0")

{

myOut\_Class.R0\_Bit(Type.Substring(3, 1), Value);

}

if (Type.Substring(0, 2) == "R1")

{

myOut\_Class.R1\_Bit(Type.Substring(3, 1), Value);

}

if (Type.Substring(0, 2) == "R2")

{

myOut\_Class.R2\_Bit(Type.Substring(3, 1), Value);

}

if (Type.Substring(0, 2) == "R3")

{

myOut\_Class.R3\_Bit(Type.Substring(3, 1), Value);

}

if (Type.Substring(0, 2) == "R4")

{

myOut\_Class.R4\_Bit(Type.Substring(3, 1), Value);

}

if (Type.Substring(0, 2) == "R5")

{

myOut\_Class.R5\_Bit(Type.Substring(3, 1), Value);

}

}

if(Type.Length == 2)

{

if (Type.Substring(0,2) == "R0")

{

myOut\_Class.R0\_Byte(Value);

}

if (Type.Substring(0, 2) == "R1")

{

myOut\_Class.R1\_Byte(Value);

}

if (Type.Substring(0, 2) == "R2")

{

myOut\_Class.R2\_Byte(Value);

}

if (Type.Substring(0, 2) == "R3")

{

myOut\_Class.R3\_Byte(Value);

}

if (Type.Substring(0, 2) == "R4")

{

myOut\_Class.R4\_Byte(Value);

}

if (Type.Substring(0, 2) == "R5")

{

myOut\_Class.R5\_Byte(Value);

}

}

}

/// <summary>

///

/// </summary>

/// <param name="Type"></param>

/// <returns></returns>

public string Read\_Output(string Type)

{

//Code here

return null;//trả rỗng về, em lập trình trong này rồi trả về sau

}

//vâng vâng

//constructor

public PLCPi() { }

}

}

**OUT\_CLASS**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Threading;

namespace PLCPiProject

{

class Out\_Class

{

IC595\_Out myIC595\_Out = new IC595\_Out();

int[] Out\_Array = { 0, 0, 0, 0, 0, 0 };

/// <summary>

/// Method xuất ngõ ra theo bit

/// </summary>

/// <param name="Chanel">bit muốn xuất ra, 0-1-2...-7</param>

/// <param name="Value">0 or 1, bật hoặc tắt</param>

public void R0\_Bit (string Chanel, int Value)

{

if (Chanel == "0")

{

if (Value == 1)

{

Out\_Array[5] =Out\_Array[5] | 0x01;

}

else

{

Out\_Array[5] = Out\_Array[5] & 0xFE;

}

}

if (Chanel == "1")

{

if(Value == 1)

{

Out\_Array[5] = Out\_Array[5] | 0x02;

}

else

{

Out\_Array[5] = Out\_Array[5] & 0xFD;

}

}

if (Chanel == "2")

{

if(Value == 1)

{

Out\_Array[5] = Out\_Array[5] | 0x04;

}

else

{

Out\_Array[5] = Out\_Array[5] & 0xFB;

}

}

if (Chanel == "3")

{

if(Value == 1)

{

Out\_Array[5] = Out\_Array[5] | 0x08;

}

else

{

Out\_Array[5] = Out\_Array[5] & 0xF7;

}

}

if (Chanel == "4")

{

if (Value == 1)

{

Out\_Array[5] = Out\_Array[5] | 0x10;

}

else

{

Out\_Array[5] = Out\_Array[5] & 0xEF;

}

}

if (Chanel == "5")

{

if(Value == 1)

{

Out\_Array[5] = Out\_Array[5] | 0x20;

}

else

{

Out\_Array[5] = Out\_Array[5] & 0xDF;

}

}

if (Chanel == "6")

{

if(Value == 1)

{

Out\_Array[5] = Out\_Array[5] | 0x40;

}

else

{

Out\_Array[5] = Out\_Array[5] & 0xBF;

}

}

if (Chanel == "7")

{

if(Value == 1)

{

Out\_Array[5] = Out\_Array[5] | 0x80;

}

else

{

Out\_Array[5] = Out\_Array[5] & 0x7F;

}

}

myIC595\_Out.DichDuLieu(Out\_Array);

}

/// <summary>

/// Method xuất ngõ ra theo bit

/// </summary>

/// <param name="Chanel">bit muốn xuất ra, 0-1-2...-7</param>

/// <param name="Value">0 or 1, bật hoặc tắt</param>

public void R1\_Bit(string Chanel, int Value)

{

if (Chanel == "0")

{

if (Value == 1)

{

Out\_Array[4] =Out\_Array[4] | 0x01;

}

else

{

Out\_Array[4] = Out\_Array[4] & 0xFE;

}

}

if (Chanel == "1")

{

if(Value == 1)

{

Out\_Array[4] = Out\_Array[4] | 0x02;

}

else

{

Out\_Array[4] = Out\_Array[4] & 0xFD;

}

}

if (Chanel == "2")

{

if(Value == 1)

{

Out\_Array[4] = Out\_Array[4] | 0x04;

}

else

{

Out\_Array[4] = Out\_Array[4] & 0xFB;

}

}

if (Chanel == "3")

{

if(Value == 1)

{

Out\_Array[4] = Out\_Array[4] | 0x08;

}

else

{

Out\_Array[4] = Out\_Array[4] & 0xF7;

}

}

if (Chanel == "4")

{

if (Value == 1)

{

Out\_Array[4] = Out\_Array[4] | 0x10;

}

else

{

Out\_Array[4] = Out\_Array[4] & 0xEF;

}

}

if (Chanel == "5")

{

if(Value == 1)

{

Out\_Array[4] = Out\_Array[4] | 0x20;

}

else

{

Out\_Array[4] = Out\_Array[4] & 0xDF;

}

}

if (Chanel == "6")

{

if(Value == 1)

{

Out\_Array[4] = Out\_Array[4] | 0x40;

}

else

{

Out\_Array[4] = Out\_Array[4] & 0xBF;

}

}

if (Chanel == "7")

{

if(Value == 1)

{

Out\_Array[4] = Out\_Array[4] | 0x80;

}

else

{

Out\_Array[4] = Out\_Array[4] & 0x7F;

}

}

myIC595\_Out.DichDuLieu(Out\_Array);

}

/// <summary>

/// Method xuất ngõ ra theo bit

/// </summary>

/// <param name="Chanel">bit muốn xuất ra, 0-1-2...-7</param>

/// <param name="Value">0 or 1, bật hoặc tắt</param>

public void R2\_Bit(string Chanel, int Value)

{

if (Chanel == "0")

{

if (Value == 1)

{

Out\_Array[3] =Out\_Array[3] | 0x01;

}

else

{

Out\_Array[3] = Out\_Array[3] & 0xFE;

}

}

if (Chanel == "1")

{

if(Value == 1)

{

Out\_Array[3] = Out\_Array[3] | 0x02;

}

else

{

Out\_Array[3] = Out\_Array[3] & 0xFD;

}

}

if (Chanel == "2")

{

if(Value == 1)

{

Out\_Array[3] = Out\_Array[3] | 0x04;

}

else

{

Out\_Array[3] = Out\_Array[3] & 0xFB;

}

}

if (Chanel == "3")

{

if(Value == 1)

{

Out\_Array[3] = Out\_Array[3] | 0x08;

}

else

{

Out\_Array[3] = Out\_Array[3] & 0xF7;

}

}

if (Chanel == "4")

{

if (Value == 1)

{

Out\_Array[3] = Out\_Array[3] | 0x10;

}

else

{

Out\_Array[3] = Out\_Array[3] & 0xEF;

}

}

if (Chanel == "5")

{

if(Value == 1)

{

Out\_Array[3] = Out\_Array[3] | 0x20;

}

else

{

Out\_Array[3] = Out\_Array[3] & 0xDF;

}

}

if (Chanel == "6")

{

if(Value == 1)

{

Out\_Array[3] = Out\_Array[3] | 0x40;

}

else

{

Out\_Array[3] = Out\_Array[3] & 0xBF;

}

}

if (Chanel == "7")

{

if(Value == 1)

{

Out\_Array[3] = Out\_Array[3] | 0x80;

}

else

{

Out\_Array[3] = Out\_Array[3] & 0x7F;

}

}

myIC595\_Out.DichDuLieu(Out\_Array);

}

/// <summary>

/// Method xuất ngõ ra theo bit

/// </summary>

/// <param name="Chanel">bit muốn xuất ra, 0-1-2...-7</param>

/// <param name="Value">0 or 1, bật hoặc tắt</param>

public void R3\_Bit(string Chanel, int Value)

{

if (Chanel == "0")

{

if (Value == 1)

{

Out\_Array[2] = Out\_Array[2] | 0x01;

}

else

{

Out\_Array[2] = Out\_Array[2] & 0xFE;

}

}

if (Chanel == "1")

{

if(Value == 1)

{

Out\_Array[2] = Out\_Array[2] | 0x02;

}

else

{

Out\_Array[2] = Out\_Array[2] & 0xFD;

}

}

if (Chanel == "2")

{

if(Value == 1)

{

Out\_Array[2] = Out\_Array[2] | 0x04;

}

else

{

Out\_Array[2] = Out\_Array[2] & 0xFB;

}

}

if (Chanel == "3")

{

if(Value == 1)

{

Out\_Array[2] = Out\_Array[2] | 0x08;

}

else

{

Out\_Array[2] = Out\_Array[2] & 0xF7;

}

}

if (Chanel == "4")

{

if (Value == 1)

{

Out\_Array[2] = Out\_Array[2] | 0x10;

}

else

{

Out\_Array[2] = Out\_Array[2] & 0xEF;

}

}

if (Chanel == "5")

{

if(Value == 1)

{

Out\_Array[2] = Out\_Array[2] | 0x20;

}

else

{

Out\_Array[2] = Out\_Array[2] & 0xDF;

}

}

if (Chanel == "6")

{

if(Value == 1)

{

Out\_Array[2] = Out\_Array[2] | 0x40;

}

else

{

Out\_Array[2] = Out\_Array[2] & 0xBF;

}

}

if (Chanel == "7")

{

if(Value == 1)

{

Out\_Array[2] = Out\_Array[2] | 0x80;

}

else

{

Out\_Array[2] = Out\_Array[2] & 0x7F;

}

}

myIC595\_Out.DichDuLieu(Out\_Array);

}

/// <summary>

/// Method xuất ngõ ra theo bit

/// </summary>

/// <param name="Chanel">bit muốn xuất ra, 0-1-2...-7</param>

/// <param name="Value">0 or 1, bật hoặc tắt</param>

public void R4\_Bit(string Chanel, int Value)

{

if (Chanel == "0")

{

if (Value == 1)

{

Out\_Array[1] = Out\_Array[1] | 0x01;

}

else

{

Out\_Array[1] = Out\_Array[1] & 0xFE;

}

}

if (Chanel == "1")

{

if(Value == 1)

{

Out\_Array[1] = Out\_Array[1] | 0x02;

}

else

{

Out\_Array[1] = Out\_Array[1] & 0xFD;

}

}

if (Chanel == "2")

{

if(Value == 1)

{

Out\_Array[1] = Out\_Array[1] | 0x04;

}

else

{

Out\_Array[1] = Out\_Array[1] & 0xFB;

}

}

if (Chanel == "3")

{

if(Value == 1)

{

Out\_Array[1] = Out\_Array[1] | 0x08;

}

else

{

Out\_Array[1] = Out\_Array[1] & 0xF7;

}

}

if (Chanel == "4")

{

if (Value == 1)

{

Out\_Array[1] = Out\_Array[1] | 0x10;

}

else

{

Out\_Array[1] = Out\_Array[1] & 0xEF;

}

}

if (Chanel == "5")

{

if(Value == 1)

{

Out\_Array[1] = Out\_Array[1] | 0x20;

}

else

{

Out\_Array[1] = Out\_Array[1] & 0xDF;

}

}

if (Chanel == "6")

{

if(Value == 1)

{

Out\_Array[1] = Out\_Array[1] | 0x40;

}

else

{

Out\_Array[1] = Out\_Array[1] & 0xBF;

}

}

if (Chanel == "7")

{

if(Value == 1)

{

Out\_Array[1] = Out\_Array[1] | 0x80;

}

else

{

Out\_Array[1] = Out\_Array[1] & 0x7F;

}

}

myIC595\_Out.DichDuLieu(Out\_Array);

}

/// <summary>

/// Method xuất ngõ ra theo bit

/// </summary>

/// <param name="Chanel">bit muốn xuất ra, 0-1-2...-7</param>

/// <param name="Value">0 or 1, bật hoặc tắt</param>

public void R5\_Bit(string Chanel, int Value)

{

if (Chanel == "0")

{

if (Value == 1)

{

Out\_Array[0] = Out\_Array[0] | 0x01;

}

else

{

Out\_Array[0] = Out\_Array[0] & 0xFE;

}

}

if (Chanel == "1")

{

if(Value == 1)

{

Out\_Array[0] = Out\_Array[0] | 0x02;

}

else

{

Out\_Array[0] = Out\_Array[0] & 0xFD;

}

}

if (Chanel == "2")

{

if(Value == 1)

{

Out\_Array[0] = Out\_Array[0] | 0x04;

}

else

{

Out\_Array[0] = Out\_Array[0] & 0xFB;

}

}

if (Chanel == "3")

{

if(Value == 1)

{

Out\_Array[0] = Out\_Array[0] | 0x08;

}

else

{

Out\_Array[0] = Out\_Array[0] & 0xF7;

}

}

if (Chanel == "4")

{

if (Value == 1)

{

Out\_Array[0] = Out\_Array[0] | 0x10;

}

else

{

Out\_Array[0] = Out\_Array[0] & 0xEF;

}

}

if (Chanel == "5")

{

if(Value == 1)

{

Out\_Array[0] = Out\_Array[0] | 0x20;

}

else

{

Out\_Array[0] = Out\_Array[0] & 0xDF;

}

}

if (Chanel == "6")

{

if(Value == 1)

{

Out\_Array[0] = Out\_Array[0] | 0x40;

}

else

{

Out\_Array[0] = Out\_Array[0] & 0xBF;

}

}

if (Chanel == "7")

{

if(Value == 1)

{

Out\_Array[0] = Out\_Array[0] | 0x80;

}

else

{

Out\_Array[0] = Out\_Array[0] & 0x7F;

}

}

myIC595\_Out.DichDuLieu(Out\_Array);

}

/// <summary>

/// xuất ngõ ra theo byte

/// </summary>

/// <param name="Data">data truyền vào</param>

public void R0\_Byte(int Data)

{

Out\_Array[5] = Data;

myIC595\_Out.DichDuLieu(Out\_Array);

}

/// <summary>

/// xuất ngõ ra theo byte

/// </summary>

/// <param name="Data">data truyền vào</param>

public void R1\_Byte(int Data)

{

Out\_Array[4] = Data;

myIC595\_Out.DichDuLieu(Out\_Array);

}

/// <summary>

/// xuất ngõ ra theo byte

/// </summary>

/// <param name="Data">data truyền vào</param>

public void R2\_Byte(int Data)

{

Out\_Array[3] = Data;

myIC595\_Out.DichDuLieu(Out\_Array);

}

/// <summary>

/// xuất ngõ ra theo byte

/// </summary>

/// <param name="Data">data truyền vào</param>

public void R3\_Byte(int Data)

{

Out\_Array[2] = Data;

myIC595\_Out.DichDuLieu(Out\_Array);

}

/// <summary>

/// xuất ngõ ra theo byte

/// </summary>

/// <param name="Data">data truyền vào</param>

public void R4\_Byte(int Data)

{

Out\_Array[1] = Data;

myIC595\_Out.DichDuLieu(Out\_Array);

}

/// <summary>

/// xuất ngõ ra theo byte

/// </summary>

/// <param name="Data">data truyền vào</param>

public void R5\_Byte(int Data)

{

Out\_Array[0] = Data;

myIC595\_Out.DichDuLieu(Out\_Array);

}

}

}

**EXP\_595**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using RaspberryPiDotNet;

using System.Threading;

using Raspberry.IO.GeneralPurpose;

namespace PLCPiProject

{

class IC595\_Out

{

GPIOFile data = new GPIOFile(GPIOPins.V2\_Pin\_P1\_11, GPIODirection.Out);

GPIOFile Xuat = new GPIOFile(GPIOPins.V2\_Pin\_P1\_13, GPIODirection.Out);

GPIOFile Dich = new GPIOFile(GPIOPins.V2\_Pin\_P1\_15, GPIODirection.Out);

var data = ConnectorPin.P1Pin16.ToProcessor();

var dich = ConnectorPin.P1Pin22.ToProcessor();

var xuat = ConnectorPin.P1Pin18.ToProcessor();

byte a = 0x80;

//Method export595

//Dat lun bang tieng viet di

//vi đối tượng khách hàng của mình ở đây là người việt

/// <summary>

/// dịch dữ liệu vào 595 để xuất ra ngõ ra

/// </summary>

/// <param name="DuLieu">mảng 6 phần tử, chứa dữ liệu cần xuất ra</param>

public void DichDuLieu(int[] DuLieu)

{

for (int j = 0; j <6; j++)

{

for (int i = 0; i < 8; i++)

{

if ((DuLieu[j] & a) == 0x80)

{

data.Write(PinState.High);

//Console.WriteLine("11");

}

else

{

data.Write(PinState.Low);

//Console.WriteLine("00");

}

Dich.Write(PinState.Low);

Dich.Write(PinState.High);

DuLieu[j] <<= 1;

}

}

Xuat.Write(PinState.Low);

Xuat.Write(PinState.High);

}

//còn có thể có nhiều method hoặc properties khác, phù hợp với yêu cầu của em

//Constructor

//nếu ko cần khởi tạo gì đầu tiên thì ko cần khai báo method constructor này

//vì hướng dẫn nên anh cứ tạo nhưng ko bỏ gì vào

public IC595\_Out() { }

}

}

**IN\_CLASS**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Threading;

using RaspberryPiDotNet;

namespace PLCPiProject

{

class In\_Class

{

GPIOFile Add10 = new GPIOFile(GPIOPins.V2Plus\_Pin\_P1\_29, GPIODirection.Out);

GPIOFile Add11 = new GPIOFile(GPIOPins.V2Plus\_Pin\_P1\_31, GPIODirection.Out);

GPIOFile Add12 = new GPIOFile(GPIOPins.V2Plus\_Pin\_P1\_33, GPIODirection.Out);

GPIOFile Add20 = new GPIOFile(GPIOPins.V2Plus\_Pin\_P1\_36, GPIODirection.Out);

GPIOFile Add21 = new GPIOFile(GPIOPins.V2Plus\_Pin\_P1\_38, GPIODirection.Out);

GPIOFile Add22 = new GPIOFile(GPIOPins.V2Plus\_Pin\_P1\_40, GPIODirection.Out);

GPIOFile EnableV0 = new GPIOFile(GPIOPins.V2\_Pin\_P1\_23, GPIODirection.Out);

GPIOFile EnableV1 = new GPIOFile(GPIOPins.V2\_Pin\_P1\_19, GPIODirection.Out);

GPIOFile EnableV2 = new GPIOFile(GPIOPins.V2\_Pin\_P1\_21, GPIODirection.Out);

GPIOFile EnableV3 = new GPIOFile(GPIOPins.V2\_Pin\_P1\_26, GPIODirection.Out);

GPIOFile EnableV4 = new GPIOFile(GPIOPins.V2\_Pin\_P1\_24, GPIODirection.Out);

GPIOFile In\_151 = new GPIOFile(GPIOPins.V2Plus\_Pin\_P1\_35, GPIODirection.In);

GPIOFile In\_152 = new GPIOFile(GPIOPins.V2Plus\_Pin\_P1\_32, GPIODirection.In);

int[] In\_Array = { 0, 0, 0, 0, 0, 0, 0, 0 };

public Int16 V0\_Bit(string Chanel)

{

EnableV0.Write(PinState.Low);

EnableV1.Write(PinState.High);

EnableV2.Write(PinState.High);

if (Chanel == "0")

{

Add10.Write(PinState.Low);

Add11.Write(PinState.Low);

Add12.Write(PinState.Low);

In\_Array[7] = In\_151.Read();

return In\_Array[7];

}

}

}

}