

ภาคผนวก ข

โปรแกรมควบคุมการทำงานของรถขนส่งและลำเลียงอัตโนมัติ (AGV)

โปรแกรมควบคุมการทำงานของรถ AGV และ Server

```

using System; //แหล่งที่เก็บคลาสของกลุ่ม System
using System.Collections.Generic; //แหล่งที่เก็บคลาสของกลุ่ม
System.Collections.Generic
using System.ComponentModel; //แหล่งที่เก็บคลาสของกลุ่ม
System.ComponentModel
using System.Data; //แหล่งที่เก็บคลาสของกลุ่ม System.Data
using System.Drawing; //แหล่งที่เก็บคลาสของกลุ่ม System.Drawing
using System.Linq; //แหล่งที่เก็บคลาสของกลุ่ม System.Linq
using System.Text; //แหล่งที่เก็บคลาสของกลุ่ม System.Text
using System.Windows.Forms; //แหล่งที่เก็บคลาสของกลุ่ม System.Windows.Forms
using System.Net; //แหล่งที่เก็บคลาสของกลุ่ม Net
using System.Net.Sockets; //แหล่งที่เก็บคลาสของกลุ่ม System.Net.Sockets
using System.Collections; //แหล่งที่เก็บคลาสของกลุ่ม System.Collections
using System.IO.Ports; //แหล่งที่เก็บคลาสของกลุ่ม System.IO.Ports
using System.Threading; //แหล่งที่เก็บคลาสของกลุ่ม System.Threading

namespace WindowsFormsApplication3
{
    public partial class Form1 : Form
    {
        #region Drive //กำหนดตัวพื้นที่เฉพาะการความมอเตอร์ เรียกว่า region Drive
        SerialPort sp1, sp2; //กำหนดตัวแปรของ SerialPort เป็นตัวแปร sp1 และ sp2
        string inData1, inData2; //กำหนดตัวแปรประเภท string ชื่อว่า InData1 และ
        InData2

        public Form1()
        {
            InitializeComponent();
            InitializeSerialComponent();
        }

        private void InitializeSerialComponent()
        {
            try
            {
                sp1 = new SerialPort("COM8", 115200, Parity.None, 8, StopBits.One); //
                กำหนดค่าการทำงานของ SerialPort1
                sp2 = new SerialPort("COM6", 9600, Parity.None, 8, StopBits.One); //
                กำหนดค่าการทำงานของ SerialPort2
            }
            catch { }
        }
    }
}

```

```

sp1.Open();//กำหนดให้ SerialPort1 เริ่มทำงาน
sp2.Open();//กำหนดให้ SerialPort2 เริ่มทำงาน

sp1.Write("1");//กำหนดให้ SerialPort1 เขียนเลข 1
sp2.Write("\\\\\\\\\\\\\\\\\\\\COM8\r");//กำหนดให้ SerialPort2 เขียน
\\\\\\\\\\\\\\\\\\\\\\COM8
    }
}

// Serial Port 1 Received Data
private delegate void SetTextRichTextControl1();

void sp1_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    {
        inData1 = sp1.ReadExisting(); // กำหนดให้ค่าของ inData1 =
sp1.ReadExisting()
        {
            bbb += inData1[i]; //ค่าของ bbb จะเท่ากับค่าของ inData ในรูปของตัวแปร
Array
        }
        calculate();//ไปยังฟังก์ชัน calculate()
    }
}

void sp2_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    inData2 = sp2.ReadExisting();// กำหนดให้ค่าของ inData2 =
sp2.ReadExisting()

}

private void calculate()
{
    if (textBox2.Text == "1111111111111111")//ถ้าค่าใน textBox2.Text =
1111111111111111 ให้ทำตามเงื่อนไขด้านล่างนี้
    {
        sp2.Write("!M 0 0");//ให้ SerialPort2 เขียน !M 0 0 ออกไป
        return; //ออกจากลูปการทำงาน
    }
}

```

else if (textBox2.Text == "0000000000000000") ถ้าค่าใน textBox2.Text = 0000000000000000 ให้ทำตามเงื่อนไขด้านล่างนี้

```
{
    sp2.Write("!M 0 0");//ให้ SerialPort2 เขียน !M 0 0 ออกไป
    return; //ออกจากลูปการทำงาน
}
else if (textBox3.Text == "1")
{
    // "0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15" Read sensor bit 0 - 15
    BeginInvoke(new EventHandler(delegate
    {
        if (textBox2.Text == "0000000000000111") sp2.Write("!M -50 -
300");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -50 -300
ออกไป

        else if (textBox2.Text == "0000000000001111") sp2.Write("!M -80 -
300");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -80 -300
ออกไป

        else if (textBox2.Text == "0000000000011111") sp2.Write("!M -100
-300");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -100 -300
ออกไป

        else if (textBox2.Text == "0000000000111111") sp2.Write("!M -145
-300");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -145 -300
ออกไป

        else if (textBox2.Text == "0000000000111110") sp2.Write("!M -145
-300");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -145 -300
ออกไป

        else if (textBox2.Text == "0000000001111110") sp2.Write("!M -155
-300");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -155 -300
ออกไป

        else if (textBox2.Text == "0000000001111100") sp2.Write("!M -180
-300");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -180 -300
ออกไป

        else if (textBox2.Text == "0000000011111100") sp2.Write("!M -200
-300");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -200 -300
ออกไป

        else if (textBox2.Text == "0000000011111000") sp2.Write("!M -210
-300");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -220 -300
ออกไป
```

```

else if (textBox2.Text == "0000000111111000") sp2.Write("!M -240
-300");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -240 -300
ออกไป

else if (textBox2.Text == "0000000111110000") sp2.Write("!M -260
-300");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -260 -300
ออกไป

else if (textBox2.Text == "0000001111110000") sp2.Write("!M -280
-300");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -280 -300
ออกไป

else if (textBox2.Text == "0000001111100000") sp2.Write("!M -300
-300");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -300 -300
ออกไป

else if (textBox2.Text == "0000011111100000") sp2.Write("!M -300
-300");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -300 -300
ออกไป

else if (textBox2.Text == "0000011111000000") sp2.Write("!M -300
-300");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -300 -300
ออกไป

else if (textBox2.Text == "0000111111000000") sp2.Write("!M -300
-280");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -300 -280
ออกไป

else if (textBox2.Text == "0000111110000000") sp2.Write("!M -300
-260");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -300 -260
ออกไป

else if (textBox2.Text == "0001111110000000") sp2.Write("!M -300
-240");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -300 -240
ออกไป

else if (textBox2.Text == "0001111100000000") sp2.Write("!M -300
-210");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -300 -210
ออกไป

else if (textBox2.Text == "0011111100000000") sp2.Write("!M -300
-200");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -300 -200
ออกไป

else if (textBox2.Text == "0011111000000000") sp2.Write("!M -300
-180");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -300 -180
ออกไป

```

```

        else if (textBox2.Text == "0111111000000000") sp2.Write("!M -300
-155");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -300 -155
ออกไป

        else if (textBox2.Text == "0111110000000000") sp2.Write("!M -300
-155");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -300 -155
ออกไป

        else if (textBox2.Text == "1111110000000000") sp2.Write("!M -300
-145");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -300 -145
ออกไป

        else if (textBox2.Text == "1111100000000000") sp2.Write("!M -300
-100");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -300 -100
ออกไป

        else if (textBox2.Text == "1111000000000000") sp2.Write("!M -300
-80");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -300 -80
ออกไป
        else if (textBox2.Text == "1110000000000000") sp2.Write("!M -300
-50");//ถ้าค่าใน textBox2.Text ตรงตามเงื่อนไขให้ SerialPort2 เขียน !M -300 -80
ออกไป
    });
}
private void button3_Click(object sender, EventArgs e)
{
    if (sp2.IsOpen) //ถ้า SerialPort2 เชื่อมต่ออยู่ให้ทำตามเงื่อนไขข้างใต้
    {
        if (this.button2.BackColor == System.Drawing.SystemColors.Control)
        {
            sp2.Write("!EX" + '\r');//ให้ SerialPort2 เขียน !EX
            this.button2.BackColor = System.Drawing.Color.Red;
        }
        else
        {
            sp2.Write("!MG" + '\r');//ให้ SerialPort2 เขียน !MG
            this.button2.BackColor = System.Drawing.SystemColors.Control;
        }
    }
}
}
#endregion

////////////////////////////////////
//

```

```

#region TCP/IP //กำหนดส่วนของ TCP/IP

String GetIP()//ทำการรับค่า IP
{
    String strHostName = Dns.GetHostName();

    // Find host by name
    IPEndPoint iphostentry = Dns.GetHostByName(strHostName);

    // Grab the first IP addresses
    String IPStr = "";
    foreach (IPAddress ipaddress in iphostentry.AddressList)
    {
        IPStr = ipaddress.ToString();
        return IPStr;
    }
}

private void buttonStopListen_Click(object sender, EventArgs e) //กำหนดปุ่ม
หยุดการเชื่อมต่อ
{
    CloseSockets();//กำหนดให้ทำการปิดข้อกเกิดการเชื่อมต่อ
}

private void UpdateControls(bool listening) //กำหนดส่วนของการอัปเดตสถานะของ
การเชื่อมต่อ
{
    buttonStartListen.Enabled = !listening; //ถ้าปุ่มเริ่มการทำงานสามารถกดได้
แสดงว่า การเชื่อมต่อั้นยังไม่มีมีการเชื่อมต่ออยู่
    buttonStopListen.Enabled = listening; //ถ้าปุ่มหยุดการทำงานสามารถกดได้แสดง
ว่า การเชื่อมต่อั้นยังมีการเชื่อมต่ออยู่
}

// This is the call back function, which will be invoked when a client is
connected
public void OnClientConnect(IAsyncResult asyn)
{
    try
    {
        // Here we complete/end the BeginAccept() asynchronous call
        // by calling EndAccept() - which returns the reference to
        // a new Socket object
        Socket workerSocket = m_mainSocket.EndAccept(asyn);
    }
}

```

```

// Now increment the client count for this client
// in a thread safe manner
Interlocked.Increment(ref m_clientCount);

// Add the workerSocket reference to our ArrayList
m_workerSocketList.Add(workerSocket);

// Send a welcome message to client
string msg = "Position Currently" + textBox5.Text + "\n";
SendMsgToClient(msg,m_clientCount);

// Update the list box showing the list of clients (thread safe call)
UpdateClientListControl();

// Let the worker Socket do the further processing for the
// just connected client
WaitForData(workerSocket, m_clientCount);

// Since the main Socket is now free, it can go back and wait for
// other clients who are attempting to connect
m_mainSocket.BeginAccept(new AsyncCallback(OnClientConnect), null);
}

}

void SendMsgToClient(string msg, int clientNumber)
{
    // Convert the reply to byte array
    byte[] byData = System.Text.Encoding.ASCII.GetBytes(msg);

    Socket workerSocket = (Socket)m_workerSocketList[clientNumber - 1];
    workerSocket.Send(byData); //ทำการกำหนดหมายเลขของตัว Client ที่ทำการ
    เชื่อมต่อ
}

private void UpdateClientListControl()
{
    if (InvokeRequired) // Is this called from a thread other than the one
    created
    // the control
    {
        // We cannot update the GUI on this thread.
        // All GUI controls are to be updated by the main (GUI) thread.
        // Hence we will use the invoke method on the control which will
        // be called when the Main thread is free
        // Do UI update on UI thread
        listBoxClientList.BeginInvoke(new
        UpdateClientListCallback(UpdateClientList), null);
    }
}

```

```

    }
    else
    {
        // This is the main thread which created this control, hence update it
        // directly
        UpdateClientList();
    }
}
public class SocketPacket
{
    // Constructor which takes a Socket and a client number
    public SocketPacket(System.Net.Sockets.Socket socket, int clientNumber)
    {
        m_currentSocket = socket;
        m_clientNumber = clientNumber;
    }
    public System.Net.Sockets.Socket m_currentSocket;
    public int m_clientNumber;
    // Buffer to store the data sent by the client
    public byte[] dataBuffer = new byte[1024];
}

// Start waiting for data from the client
// Update the list of clients that is displayed
void UpdateClientList()
{
    listBoxClientList.Items.Clear();
    for (int i = 0; i < m_workerSocketList.Count; i++)
    {
        string clientKey = Convert.ToString(i + 1);
        Socket workerSocket = (Socket)m_workerSocketList[i];
        if (workerSocket != null)
        {
            if (workerSocket.Connected)
            {
                listBoxClientList.Items.Add(clientKey);
            }
        }
    }
}
public void OnDataReceived(IAsyncResult asyn)
{
    SocketPacket socketData = (SocketPacket)asyn.AsyncState;
    {
        // Complete the BeginReceive() asynchronous call by EndReceive()
method    // which will return the number of characters written to the stream

```

```

// by the client
int iRx = socketData.m_currentSocket.EndReceive(async);
char[] chars = new char[iRx + 1];
// Extract the characters as a buffer
System.Text.Decoder d = System.Text.Encoding.UTF8.GetDecoder();
int charLen = d.GetChars(socketData.dataBuffer,
    0, iRx, chars, 0);

System.String szData = new System.String(chars);
string msg = "" + socketData.m_clientNumber + ":";
AppendToRichEditControl(msg + szData);

// Continue the waiting for data on the Socket
WaitForData(socketData.m_currentSocket,
socketData.m_clientNumber);

    }
  }
}

```

ภาคผนวก ค
โปรแกรมควบคุมการสั่งการจากสถานี

โปรแกรมควบคุมการสั่งการจากสถานี (Client)

```

using System; //แหล่งที่เก็บคลาสของกลุ่ม System
using System.Collections.Generic; //แหล่งที่เก็บคลาสของกลุ่ม
System.Collections.Generic
using System.ComponentModel; //แหล่งที่เก็บคลาสของกลุ่ม
System.ComponentModel
using System.Data; //แหล่งที่เก็บคลาสของกลุ่ม System.Data
using System.Drawing; //แหล่งที่เก็บคลาสของกลุ่ม System.Drawing
using System.Linq; //แหล่งที่เก็บคลาสของกลุ่ม System.Linq
using System.Text; //แหล่งที่เก็บคลาสของกลุ่ม System.Text
using System.Windows.Forms; //แหล่งที่เก็บคลาสของกลุ่ม System.Windows.Forms
using System.Net; //แหล่งที่เก็บคลาสของกลุ่ม Net
using System.Net.Sockets; //แหล่งที่เก็บคลาสของกลุ่ม System.Net.Sockets

namespace SocketClient
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent(); //เรียกค่าเริ่มต้นของหน้าต่างโปรแกรมออกมา
            textBoxIP.Text = GetIP(); //ทำการรับค่า IP ของเครื่องเข้ามายังกล่องข้อความ
        }

        String GetIP()
        {
            String strHostName = Dns.GetHostName();

            // Find host by name
            IPHostEntry iphostentry = Dns.GetHostByName(strHostName);

            // Grab the first IP addresses
            String IPStr = "";
            foreach (IPAddress ipaddress in iphostentry.AddressList)
            {
                IPStr = ipaddress.ToString();
                return IPStr;
            }
            return IPStr;
        }

        private void buttonConnect_Click(object sender, EventArgs e)

```

```

{
    // See if we have text on the IP and Port text fields
    if (textBoxIP.Text == "" || textBoxPort.Text == "")
    {
        MessageBox.Show("IP Address and Port Number are required to connect
to the Server\n");
        return;
    }
    try
    {
        UpdateControls(false);
        // Create the socket instance
        m_clientSocket = new Socket(AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.Tcp);

        // Get the remote IP address
        IPAddress ip = IPAddress.Parse(textBoxIP.Text);
        int iPortNo = System.Convert.ToInt16(textBoxPort.Text);
        // Create the end point
        IPEndPoint ipEnd = new IPEndPoint(ip, iPortNo);
        // Connect to the remote host
        m_clientSocket.Connect(ipEnd);
        if (m_clientSocket.Connected)
        {
            UpdateControls(true);
            //Wait for data asynchronously
            WaitForData();
        }
    }
}

private void UpdateControls(bool connected)
{
    buttonConnect.Enabled = !connected; //ถ้าปุ่มเริ่มการทำงานสามารถกดได้ แสดง
ว่า การเชื่อมต่อั้นยังไม่มี การเชื่อมต่ออยู่
    buttonDisconnect.Enabled = connected; //ถ้าปุ่มหยุดการทำงานสามารถกดได้
แสดงว่า การเชื่อมต่อั้นยังมีการเชื่อมต่ออยู่
}

private void buttonDisconnect_Click(object sender, EventArgs e) //ปุ่มหยุด
การเชื่อมต่อ
{
    if (m_clientSocket != null) //ถ้าจำนวนของ Client ต้องไม่เท่ากับ ว่างเปล่า
    {

```

```

        m_clientSocket.Close();//ให้ Client ตัวที่กำลังเชื่อมต่ออยู่นี้ทำการหยุดการ
เชื่อมต่อ
    }
}
private void buttonSendMessage_Click(object sender, EventArgs e)
{
    NetworkStream networkStream = new NetworkStream(m_clientSocket);
    //ให้ networkStream เท่ากับหมายเลขของ Client ที่กำลังเชื่อมต่ออยู่
    streamWriter.WriteLine("1");// จากนั้นให้ทำการส่ง 1 ออกไปเพื่อเป็นการบอก
ว่าสถานะที่ 1 ต้องการ
}
}
}

```