**Programación del entorno del manejo**

```csharp
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        private SerialPort Puerto;
        private string[] Puertos;
        public Form1()
        {
            InitializeComponent();
            listarPuertos();
        }

        #region STATUS
        private void defineStatus(bool status)
        {
            if (status == true)
            {
                statusLabel.Text = "Conectado";
                statusLabel.ForeColor = Color.Green;
            }
            else
            {
                statusLabel.Text = "Desconectado";
                statusLabel.ForeColor = Color.Red;
            }
        }
        #endregion

        #region LISTAR COM EN COMBOBOX
        private void listarPuertos()
        {
            Puertos = SerialPort.GetPortNames();
```

```csharp
            if (Puertos.Length > 0)
            {
                comboBoxPuertos.Items.Clear();
                foreach (string s in SerialPort.GetPortNames())
                {
                    comboBoxPuertos.Items.Add(s);
                }
                defineStatus(false);
                habilitar(true, true, false, false, false, false, false);
            }
        }
        #endregion

        #region HABILITACIONES
        private void habilitar(bool COM, bool conectar, bool desconectar, bool automovil, bool
camara, bool brazo, bool Luz)
        {
            comboBoxPuertos.Enabled = COM;
            botonConectar.Enabled = conectar;
            botonDesconectar.Enabled = desconectar;
            groupBoxAuto.Enabled = automovil;
            groupBoxCamara.Enabled = camara;
            groupBoxBrazo.Enabled = brazo;
            groupBoxLuces.Enabled = Luz;
        }
        #endregion

        #region CERRAR PUERTO
        private void cerrarPuerto()
        {
            if (Puerto != null && Puerto.IsOpen)
            {
                Puerto.Close();
                Puerto = null;
                defineStatus(false);
                listarPuertos();
            }
        }
        #endregion

        #region Boton DESCONECTAR
        private void botonDesconectar_Click(object sender, EventArgs e)
        {
```

```csharp
        cerrarPuerto();
    }
    #endregion

    #region Boton CONECTAR
    private void botonConectar_Click(object sender, EventArgs e)
    {
        try
        {
            if (comboBoxPuertos.SelectedIndex != -1)
            {
                Puerto = new SerialPort();
                Puerto.PortName = comboBoxPuertos.SelectedItem.ToString();
                Puerto.BaudRate = 9600;
            }
            else
            {
                throw new Exception("Seleccione un puerto para conectar");
            }
            if (!Puerto.IsOpen)
            {
                Puerto.Open();
                defineStatus(true);
                habilitar(false, false, true, true, true, true, true);
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(this, "NO ES POSIBLE LA CONEXION. \n" + ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
    #endregion

    #region Cerrar Formulario
    private void Form1_FormClosed(object sender, FormClosedEventArgs e)
    {
        cerrarPuerto();
    }
    #endregion

    #region BOTONES AUTO
    private void btUpAuto_Click(object sender, EventArgs e)
```

```csharp
{
    byte[] mBuffer = new byte[1];
    mBuffer[0] = 0x38; //Letra 8 en ascii
    Puerto.Write(mBuffer, 0, mBuffer.Length);
    labelMovAuto.Text = "Adelante...";
}

private void btDownAuto_Click(object sender, EventArgs e)
{
    byte[] mBuffer = new byte[1];
    mBuffer[0] = 0x35; //Letra 5 en ascii
    Puerto.Write(mBuffer, 0, mBuffer.Length);
    labelMovAuto.Text = "Atras...";
}

private void btIzqAuto_Click(object sender, EventArgs e)
{
    byte[] mBuffer = new byte[1];
    mBuffer[0] = 0x34; //Letra 4 en ascii
    Puerto.Write(mBuffer, 0, mBuffer.Length);
    labelMovAuto.Text = "Izquierda...";
}

private void btDerAuto_Click(object sender, EventArgs e)
{
    byte[] mBuffer = new byte[1];
    mBuffer[0] = 0x36; //Letra 6 en ascii
    Puerto.Write(mBuffer, 0, mBuffer.Length);
    labelMovAuto.Text = "Derecha...";
}
#endregion

#region BOTONES CAMARA
private void btIzqCamara_Click(object sender, EventArgs e)
{
    byte[] mBuffer = new byte[1];
    mBuffer[0] = 0x4A; //Letra J en ascii
    Puerto.Write(mBuffer, 0, mBuffer.Length);
    labelMovCam.Text = "Izquierda...";
}

private void btDerCamara_Click(object sender, EventArgs e)
{
```

```csharp
    byte[] mBuffer = new byte[1];
    mBuffer[0] = 0x4C; //Letra L en ascii
    Puerto.Write(mBuffer, 0, mBuffer.Length);
    labelMovCam.Text = "Derecha...";
}

private void btUpCamara_Click(object sender, EventArgs e)
{
    byte[] mBuffer = new byte[1];
    mBuffer[0] = 0x49; //Letra I en ascii
    Puerto.Write(mBuffer, 0, mBuffer.Length);
    labelMovCam.Text = "Arriba...";
}

private void btDownCamara_Click(object sender, EventArgs e)
{
    byte[] mBuffer = new byte[1];
    mBuffer[0] = 0x4B; //Letra K en ascii
    Puerto.Write(mBuffer, 0, mBuffer.Length);
    labelMovCam.Text = "Abajo...";
}
#endregion

#region BOTONES BRAZO
private void btAntiBrazo_Click(object sender, EventArgs e)
{
    byte[] mBuffer = new byte[1];
    mBuffer[0] = 0x41; //Letra A en ascii
    Puerto.Write(mBuffer, 0, mBuffer.Length);
    labelMovBrazo.Text = "Antihorario...";
}

private void btHoraBrazo_Click(object sender, EventArgs e)
{
    byte[] mBuffer = new byte[1];
    mBuffer[0] = 0x44; //Letra D en ascii
    Puerto.Write(mBuffer, 0, mBuffer.Length);
    labelMovBrazo.Text = "Horario...";
}

private void btSubirBrazo_Click(object sender, EventArgs e)
{
    byte[] mBuffer = new byte[1];
```

```csharp
    mBuffer[0] = 0x57; //Letra W en ascii
    Puerto.Write(mBuffer, 0, mBuffer.Length);
    labelMovBrazo.Text = "Subiendo...";
}

private void btBajarBrazo_Click(object sender, EventArgs e)
{
    byte[] mBuffer = new byte[1];
    mBuffer[0] = 0x53; //Letra S en ascii
    Puerto.Write(mBuffer, 0, mBuffer.Length);
    labelMovBrazo.Text = "Bajando...";
}

private void btAbrirPinza_Click(object sender, EventArgs e)
{
    byte[] mBuffer = new byte[1];
    mBuffer[0] = 0x5A; //Letra Z en ascii
    Puerto.Write(mBuffer, 0, mBuffer.Length);
    labelMovBrazo.Text = "Abriendo...";
}

private void btCerrarPinza_Click(object sender, EventArgs e)
{
    byte[] mBuffer = new byte[1];
    mBuffer[0] = 0x58; //Letra X en ascii
    Puerto.Write(mBuffer, 0, mBuffer.Length);
    labelMovBrazo.Text = "Cerrando...";
}
#endregion

#region TECLAS
protected override bool ProcessCmdKey(ref Message m, Keys keyData)
{
    bool blnProcess = false;
    #region TECLAS PARA AUTO
    if (keyData == Keys.Up)
    {
        blnProcess = true;
        byte[] mBuffer = new byte[1];
        mBuffer[0] = 0x38; //Letra 8 en ascii
        Puerto.Write(mBuffer, 0, mBuffer.Length);
        labelMovAuto.Text = "Adelante...";
    }
```

```csharp
if (keyData == Keys.Down)
{
    blnProcess = true;
    byte[] mBuffer = new byte[1];
    mBuffer[0] = 0x35; //Letra 5 en ascii
    Puerto.Write(mBuffer, 0, mBuffer.Length);
    labelMovAuto.Text = "Atras...";
}

if (keyData == Keys.Left)
{
    blnProcess = true;
    byte[] mBuffer = new byte[1];
    mBuffer[0] = 0x34; //Letra 4 en ascii
    Puerto.Write(mBuffer, 0, mBuffer.Length);
    labelMovAuto.Text = "Izquierda...";
}

if (keyData == Keys.Right)
{
    blnProcess = true;
    byte[] mBuffer = new byte[1];
    mBuffer[0] = 0x36; //Letra 6 en ascii
    Puerto.Write(mBuffer, 0, mBuffer.Length);
    labelMovAuto.Text = "Derecha...";
}
#endregion

#region TECLAS PARA CAMARA
if (keyData == Keys.J)
{
    blnProcess = true;
    byte[] mBuffer = new byte[1];
    mBuffer[0] = 0x4A; //Letra J en ascii
    Puerto.Write(mBuffer, 0, mBuffer.Length);
    labelMovCam.Text = "Izquierda...";
}

if (keyData == Keys.L)
{
    blnProcess = true;
    byte[] mBuffer = new byte[1];
```

```csharp
        mBuffer[0] = 0x4C; //Letra L en ascii
        Puerto.Write(mBuffer, 0, mBuffer.Length);
        labelMovCam.Text = "Derecha...";
    }

    if (keyData == Keys.I)
    {
        blnProcess = true;
        byte[] mBuffer = new byte[1];
        mBuffer[0] = 0x49; //Letra I en ascii
        Puerto.Write(mBuffer, 0, mBuffer.Length);
        labelMovCam.Text = "Arriba...";
    }

    if (keyData == Keys.K)
    {
        blnProcess = true;
        byte[] mBuffer = new byte[1];
        mBuffer[0] = 0x4B; //Letra K en ascii
        Puerto.Write(mBuffer, 0, mBuffer.Length);
        labelMovCam.Text = "Abajo...";
    }
    #endregion

    #region TECLAS PARA BRAZO
    if (keyData == Keys.D)
    {
        blnProcess = true;
        byte[] mBuffer = new byte[1];
        mBuffer[0] = 0x44; //Letra D en ascii
        Puerto.Write(mBuffer, 0, mBuffer.Length);
        labelMovBrazo.Text = "Horario...";
    }

    if (keyData == Keys.A)
    {
        blnProcess = true;
        byte[] mBuffer = new byte[1];
        mBuffer[0] = 0x41; //Letra A en ascii
        Puerto.Write(mBuffer, 0, mBuffer.Length);
        labelMovBrazo.Text = "Antihorario...";
    }
```

```csharp
if (keyData == Keys.W)
{
    blnProcess = true;
    byte[] mBuffer = new byte[1];
    mBuffer[0] = 0x57; //Letra W en ascii
    Puerto.Write(mBuffer, 0, mBuffer.Length);
    labelMovBrazo.Text = "Subiendo...";
}

if (keyData == Keys.S)
{
    blnProcess = true;
    byte[] mBuffer = new byte[1];
    mBuffer[0] = 0x53; //Letra S en ascii
    Puerto.Write(mBuffer, 0, mBuffer.Length);
    labelMovBrazo.Text = "Bajando...";
}

if (keyData == Keys.Z)
{
    blnProcess = true;
    byte[] mBuffer = new byte[1];
    mBuffer[0] = 0x5A; //Letra Z en ascii
    Puerto.Write(mBuffer, 0, mBuffer.Length);
    labelMovBrazo.Text = "Abriendo...";
}

if (keyData == Keys.X)
{
    blnProcess = true;
    byte[] mBuffer = new byte[1];
    mBuffer[0] = 0x58; //Letra X en ascii
    Puerto.Write(mBuffer, 0, mBuffer.Length);
    labelMovBrazo.Text = "Cerrando...";
}
#endregion
if (blnProcess == true)
{
    return true;
}
else
{
    return base.ProcessCmdKey(ref m, keyData);
```

```csharp
        }
    }
    #endregion

    private void btON_Click(object sender, EventArgs e)
    {
        byte[] mBuffer = new byte[1];
        mBuffer[0] = 0x47; //Letra G en ascii
        Puerto.Write(mBuffer, 0, mBuffer.Length);
    }

    private void btOFF_Click(object sender, EventArgs e)
    {
        byte[] mBuffer = new byte[1];
        mBuffer[0] = 0x42; //Letra B en ascii
        Puerto.Write(mBuffer, 0, mBuffer.Length);
    }


    }
}
```