

Proyecto: Bote de basura

En la línea del 20 al 31, abrimos un puerto serie para recibir información del Arduino

De la línea 35 a la línea 45, limpiamos los datos recibidos para imprimirlo en la interfaz

De la línea 52 en adelante, condicionamos dependiendo del número que reciba, para encender o apagar los botones que indiquen cuanta basura hay

```
10 using System;
11
12 namespace Bote_de_basura
13 {
14     3 referencias
15     public partial class Form1 : Form
16     {
17         1 referencia
18         string Recibidos;
19         public Form1()
20         {
21             InitializeComponent();
22             try
23             {
24                 if (!serialPort1.IsOpen)
25                 {
26                     serialPort1.Open();
27                 }
28                 serialPort1.DataReceived += new SerialDataReceivedEventHandler(Recepcion);
29             }
30             catch (Exception ex)
31             {
32                 MessageBox.Show("Error al abrir el puerto: " + ex.Message);
33             }
34         }
35
36         1 referencia
37         private void Recepcion(object sender, SerialDataReceivedEventArgs e)
38         {
39             string data = serialPort1.ReadExisting().Trim(); // Leer y limpiar los datos recibidos
40             // Verificar si se recibió un solo carácter
41             if (data.Length > 0)
42             {
43                 char receivedChar = data[0]; // Solo toma el primer carácter recibido
44                 // Actualizar la interfaz de usuario en el hilo principal
45                 this.Invoke(new Action(() => Actualizar(receivedChar)));
46             }
47         }
48
49         private void Actualizar(char receivedChar)
50         {
51             switch (receivedChar)
52             {
53                 // 100
54                 case '0':
55                     button1.BackColor = Color.Black;
56                     button2.BackColor = Color.Black;
57                     button3.BackColor = Color.Black;
58                     button4.BackColor = Color.Black;
59                     break;
60                 case '1':
61                     button1.BackColor = Color.White;
62                     break;
63                 // 75
64                 case '2':
65                     button2.BackColor = Color.Black;
66                     button3.BackColor = Color.Black;
67                     button4.BackColor = Color.Black;
68                     break;
69                 case '3':
70                     button2.BackColor = Color.White;
71                     break;
72                 // 50
73                 case '4':
74                     button3.BackColor = Color.Black;
75                     button4.BackColor = Color.Black;
76                     break;
77                 case '5':
78                     button3.BackColor = Color.White;
79                     break;
80                 // 25
81                 case '6':
82                     button4.BackColor = Color.Black;
83                     break;
84                 case '7':
85                     button4.BackColor = Color.White;
86                     break;
87             }
88         }
89     }
90 }
```

Código Arduino

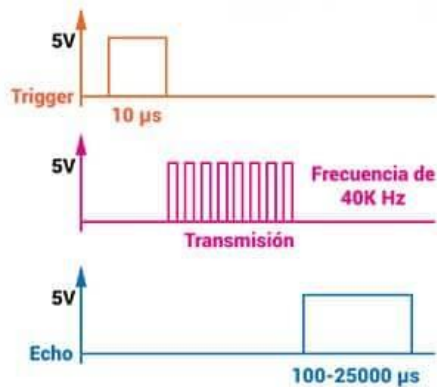
```
1 int trig = 9; // Pin para el Trigger
2 int echo = 10; // Pin para el Echo
3
4 void setup() {
5   Serial.begin(9600); // Iniciar comunicación serial
6   pinMode(trig, OUTPUT); // Establecer trig como salida
7   pinMode(echo, INPUT); // Establecer echo como entrada
8 }
9
10 void loop() {
11   long tiempo, distancia; // definimos variables para guardar el tiempo y distancia del ultrasonico
12   digitalWrite(trig, LOW);
13   delayMicroseconds(2);
14   digitalWrite(trig, HIGH);
15   delayMicroseconds(10);
16   digitalWrite(trig, LOW);
17   tiempo = pulseIn(echo, HIGH); // hasta aqui hacemos que una onda vaya y regrese, la variable tiempo sera el tiempo que tardo en ir y regresar esa onda
18   distancia = (tiempo/ 2) / 29.1; // para sacar la distancia, dividimos el tiempo que tardo en regresar la onda entre la velocidad del sonido en el aire que es 29.1
19
20   if(distancia > 0 && distancia <= 5){ // Si detectamos algo a menos de 5 cm
21     Serial.write('0'); // manda un 0 al visual
22   }else{ // de lo contrario manda un 1
23     Serial.write('1');
24   }
25   if(distancia > 5 && distancia <= 10){ // si detectamos algo a menos de 10 cm
26     Serial.write('2'); // manda un 2
27   }else{ // de lo contrario manda un 3
28     Serial.write('3');
29   }
30   if(distancia > 10 && distancia <= 15){ // si detectamos algo a menos de 15 cm
31     Serial.write('4'); // manda un 4
32   }else{ // de lo contrario manda un 5
33     Serial.write('5');
34   }
35   if(distancia > 15 && distancia <= 20){ // si detectamos algo a menos de 20 cm
36     Serial.write('6'); // manda un 6
37   }else{ // de lo contrario manda un 7
38     Serial.write('7');
39   }
40   delay(100);
41 }
```

Conexiones

PINOUT

SENSOR ULTRASONICO HC-SR04

Rango de medición
2 cm a 400 cm



Vcc 5 V
Trigger
Echo
GND

VCC conectado a 5V del Arduino

GND conectado GND del Arduino

Trig esta conectado al pin 9 del Arduino

Echo esta conectado al pin 10 del Arduino