

Proyecto: Banda transportadora

Codigo VisualBasic

Utilizamos la línea 15 para crear una variable donde abriremos un puerto a la que llamaremos Visual

Usamos las líneas 16,17 y 18 para crear variables donde se pueda enclavar los botones

En la línea 24, definiremos que la variable Visual será en el puerto COM5, esto es fundamental para la comunicación, si tu Arduino es abre otro puerto deberás de cambiar el número del COM5 a otro COM.

En la línea 25, pondremos que la velocidad de comunicación será la normal, de 9600 baudios.

Y en la línea 26, pondremos finalmente que abra el puerto de comunicación para que pueda mandar la información al Arduino.

En la línea 33, pondremos que cuando cerremos el formulario, se cierre la comunicación en Arduino.

En la línea 45, pondremos que si el botón de visual es presionado que mande una señal "1" al Arduino

En la línea 51, pondremos que si el botón de visual es presionado que mande una señal "2" al Arduino

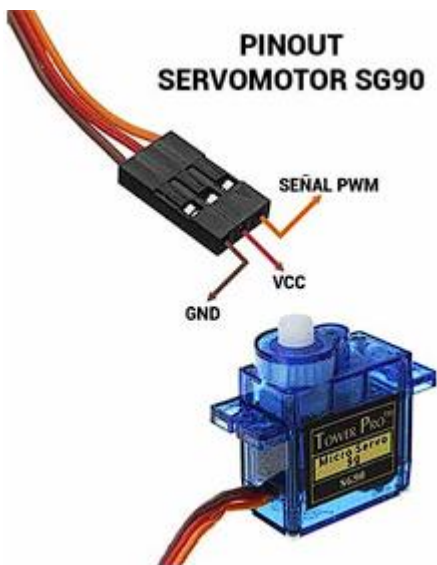
Y eso se repite en 3 botones que son los del paro de emergencia, paro y la dirección de la flecha del servomotor.

```
13 public partial class Form1 : Form
14 {
15     System.IO.Ports.SerialPort Arduino;
16     private bool btnEnclavadoPE = false;
17     private bool btnEnclavadoP = false;
18     private bool btnEnclavadoDir = false;
19
20     1 referencia
21     public Form1()
22     {
23         InitializeComponent();
24         Arduino = new System.IO.Ports.SerialPort();
25         Arduino.PortName = "COM5";
26         Arduino.BaudRate = 9600;
27         Arduino.Open();
28
29     1 referencia
30     private void Form1_FormClosing(object sender, FormClosingEventArgs e)
31     {
32         if(Arduino.IsOpen)
33         {
34             Arduino.Close();
35         }
36
37     1 referencia
38     private void Form1_Load(object sender, EventArgs e)
39     {
40
41
42     1 referencia
43     private void btnPE_Click(object sender, EventArgs e)
44     {
45         btnEnclavadoPE = !btnEnclavadoPE;
46         if (btnEnclavadoPE)
47         {
48             btnPE.Text = "Paro de emergencia DESACTIVADO";
49             Arduino.Write("1");
50     }
```

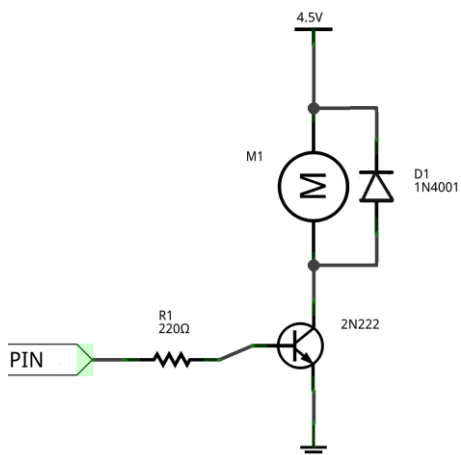
Código Arduino

```
1 #include <Servo.h> // Con esta libreria la usamos para poder usar comandos del servomotor
2 Servo myservo; // nombramos el servo como myservo
3
4 int motor = 2; // usaremos el pin 2 para accionar el motor
5 bool valorParoEmergencia; // creamos variables para hacer el paro y arranque
6 bool valorParo;
7 bool valArr;
8 bool direccionDerecho; // creamos variables para el cambio del servomotor
9 bool direccionDesvio;
10
11 int derecho = 0; // damos a la variable derecho un angulo de 0 para que quede recto el servo
12 int desvio = 70; // damos a la variable desvio un angulo de 70 para que cree un desvio
13
14 void setup() {
15     pinMode(motor, OUTPUT); // configuramos el motor como salida
16     Serial.begin(9600); // iniciamos la comunicacion con la computadora
17     myservo.attach(6); // este comando de la libreria hace referencia a que pin y que configuracion tendra el servo
18     myservo.write(0); // iniciamos el servo en el angulo 0 desde un inicio
19 }
20
21 void loop() {
22     if(Serial.available() > 0) { // condicionamos las entradas de señal leyendo lo que nos manda la computadora al arduino
23         int numero = Serial.read(); // los datos recibidos los guardamos en la variable "numero"
24         if(numero == '1') { // si la computadora nos manda un "1"
25             valorParoEmergencia = true; // indica que el paro de emergencia es presionado
26         }
27         if(numero == '2') { // si la computadora nos manda un "2"
28             valorParoEmergencia = false; // indica que el paro de emergencia no es presionado
29         }
30         if(numero == '3') { // si la computadora nos manda un "3"
31             valorParo = true; // indica que el paro es presionado
32         }
33         if(numero == '4') { // si la computadora nos manda un "4"
34             valorParo = false; // indica que el paro no es presionado
35         }
36     }
37
38     if(valorParoEmergencia == true) // si el boton de paro de emergencia esta cerrado
39     {
40         if(direccionDerecho == true) // si el boton de direccion esta presionado en derecho
41         {
42             myservo.write(derecho); // el servo estara en la posicion de derecho
43         }
44         else if(direccionDerecho == false) // si el boton de direccion está presionado en desvio
45         {
46             myservo.write(desvio); // el servo estara en la posicion de desvio
47         }
48     }
49
50     if(valArr == true && valorParo == true) // si el boton de arranque es presionado
51     {
52         digitalWrite(motor, HIGH); // inicia el motor
53     }
54     if(valorParo == false) // si el boton de paro es presionado
55     {
56         digitalWrite(motor, LOW); // el motor se para
57         myservo.write(derecho); // el servo vuelve a su estado natural
58     }
59 }
60
61 if(valorParoEmergencia == false) // si el boton de paro de emergencia esta abierto
62 {
63     digitalWrite(motor, LOW); // el motor se para
64     myservo.write(derecho); // el servo vuelve a su estado natural
65 }
66
67 delay(100);
68 }
69 }
```

Conexiones:



Café a GND del Arduino
Rojo al VCC del Arduino
Naranja al pin 6 del arduino



Pin 2 del Arduino a la base del transistor

