



Roberto Xavier Pereira Junior ▾



Cetec Capacitações

Painel > Meus cursos > Programação C# com ESP32 > 3-Comunicação Serial > Projeto Windows Forms Application (WFA) com Ambiente Gráfico

1.1.01.60 - Programação C# com ESP32 Projeto

Windows Forms Application (WFA) com Ambiente Gráfico

✓ Concluído

Crie um novo projeto WFA com nome que desejar.

Adicionar os componentes ao formulário C#

Certifique-se de que sua interface de usuário tenha os seguintes componentes:

- Um botão "Conectar" (button1) para estabelecer a conexão.
- Um botão "Desconectar" (button2) para estabelecer a conexão.
- Um Label (lblStatus) para exibir o status da conexão.
- Um checkBox (checkBox1) para ligar ou desligar o LED.
- Um comboBox, (comboBox1) para carregar as Portas COM.
- Labels de decoração.



Programando

Adicionar a variável para a Porta Serial:

```
public partial class Form1 : Form
```

```
{
```

```
    SerialPort portaSerial = new SerialPort();//adicionar esta linha
```

Duplo clique em uma área em branco do Form1, para gerar a função LOAD:

```
//laço de repetição responsável por buscar as portas COM
```

```
// e carregar o ComboBox
```

```
foreach (string porta in SerialPort.GetPortNames())
```

```
{
```

```
    comboBox1.Items.Add(porta);
```

```
}
```

```
//Propriedade para desabilitar o botão
```

```
button2.Enabled = false;
```

Duplo clique no botão Conectar:

```
if (checarESP32())
```

```
{
```

```
    //Sendo verdadeira a condição
```

```
    button1.Enabled = false; //Desabilita botão CONECTAR
```

```
    button2.Enabled = true; //Habilita botão DESCONECTAR
```

```
    comboBox1.Enabled = false; //Desabilita botão comboBox das Portas
```

```
    checkBox1.Enabled = true; //Habilita o checkBox do controle do LED
```

```
}
```

```
else
```

```
{
```

```
//Sendo falsa a condição

    button2.Enabled = false; //Desabilita botão DESCONECTAR
}
```

Duplo clique no botão Desconectar:

```
button1.Enabled = true; //Habilita o botão CONECTAR
button2.Enabled = false; //Desabilita o botão DESCONECTAR
comboBox1.SelectedIndex = -1;
comboBox1.Enabled = true;
checkBox1.Checked = false;
checkBox1.Enabled = false;
portaSerial.Write("D");
portaSerial.Close();
```

Duplo clique no checkBox1:

```
if (checkBox1.Checked)
{

    portaSerial.Write("L");
    checkBox1.Text = "Desligar Led";
}
else
{

    portaSerial.Write("D");
    checkBox1.Text = "Ligar Led";
}
```

Adicionar as funções ao código:

```
private bool checarESP32()
{
    try
    {

        //Caso algum erro inesperado ocorra neste bloco, as instruções do catch são
        executadas
    }
}
```

```

        portaSerial.PortName = comboBox1.SelectedItem.ToString(); //atribuição da
        porta selecionada

        portaSerial.BaudRate = 115200; //definição da taxa de transmissão
        portaSerial.Open(); //Abertura da conexão com a porta serial
        portaSerial.Write("TESTE"); //Envia uma string para o Arduino

        Thread.Sleep(1500); //Tempo em milissegundos para aguardar o Arduino receber,
        processar

                                //e devolver a string esperada

        string msgRecebida = portaSerial.ReadExisting(); //armazenando a resposta do
        Arduino

        //Condição para verificar se a string enviada pelo arduino é a esperada
        if (msgRecebida.Equals("TESTE"))
        {
            //Sendo a condição verdadeira, retorna true
            return true;
        }
        else
        {
            //Sendo a condição falsa
            //Verifica se a conexão com a porta serial está aberta
            if (portaSerial.IsOpen)
                portaSerial.Close(); //sendo a condição verdadeira, fecha a conexão

            return false; //retorna false
        }
    }

    catch (Exception e)
    {
        //Caso ocorra algm erro no bloco do try, uma mensagem com o erro será
        apresentada

        //E a conexão será fechada, caso esteja aberta
        MessageBox.Show(e.Message.ToString());

        if (portaSerial.IsOpen)
            portaSerial.Close();

        return false; //retorna false
    }
}

```

```
}
```

O código do ESP32 para comunicação Serial:

```
//Configurações iniciais

void setup() {

    Serial.begin(115200); //Taxa de transmissão

    pinMode(2, OUTPUT); //Definição da Porta 2 do ESP32 como saída
}


//Função principal

void loop() {

    //declaração da variável (vetor) que armazenará os
    //dados recebidos via porta serial

    char mensagem[10];

    //declaração da variável que auxiliará em indicar a
    //posição no vetor para armazenar cada caractere lido

    int i=0;

    //Condição para verificar se existe informação sendo
    //recebida pela porta serial

    if (Serial.available())

    {

        //Sendo verdadeira a condição

        //realiza a leitura do caractere e atribui no vetor

        mensagem[i] = Serial.read();

        //incrementa 1 na variável i, para que o próximo
        //caractere lido, seja armazenado na próxima posição do vetor

        i++;
    }
}
```

```
digitalWrite(2, HIGH); //Acende o Led

delay(100); //Temo de espera de 100 milissegundos

digitalWrite(2, LOW); //Apaga o Led

delay(100);

}

//adiciona o caractere que representa o final da string válida
mensagem[i] = '\0';

Serial.write(mensagem); //Envia a string recebida para a porta Serial

//Se a mensagem recebida na primeira posição do vetor

//for a letra L

if (mensagem[0] == 'L')

    digitalWrite(2, HIGH); //Liga o Led

//for a letra D

if (mensagem[0] == 'D')

    digitalWrite(2, LOW); //Apaga o led

}
```

Última atualização: terça-feira, 6 ago. 2024, 09:27

◀ Criando um Projeto Console Application com Comunicação Serial

Seguir para...




Código para o ESP32 com WebSocket ▶

CETEC CAPACITAÇÕES

Rua dos Andradas, nº 140, Santa Ifigênia - São Paulo - SP

 <https://www.cps.sp.gov.br/>

 suporte.moodlecetec@cps.sp.gov.br

 +55 11 3324-3300

