

CENTRO PAULA SOUZA

FACULDADE DE TECNOLOGIA DE JUNDIAÍ – DEPUTADO ARY FOSSEN

**MANUAL DE CONFIGURAÇÃO PARA COMUNICAÇÃO ENTRE ELIPSE E3 E
ARDUINO**

Jundiaí - SP

2018

INDICE

1. PRÉ-REQUISITOS	1
1.1. Elipse E3	1
1.2. Driver de Comunicação.....	1
1.3. Arduino.....	1
1.4. Biblioteca ModbusSlave	2
2. HARDWARE	3
3. SOFTWARE.....	3

LISTA DE FIGURAS

1. PRÉ-REQUISITOS	1
1.1. Elipse E3.....	1
Figura 1: Download Software Elipse E3.....	1
1.2. Driver de Comunicação.....	1
Figura 2: Download Driver Modbus.....	1
1.3. Arduino.....	1
Figura 3: Downlaod Arduino IDE.....	2
1.4. Biblioteca ModbusSlave	2
Figura 4: Simulando o Arduino como escravo Modbus.....	3
2. HARDWARE	3
3. SOFTWARE.....	3
Figura 5: Criando um novo domínio	4
Figura 6: Configurando o Projeto	5
Figura 7: Seguindo o assistente de aplicações.....	6
Figura 8: Aplicação configurada.....	7
Figura 9: Configurando o Driver - Modbus	8
Figura 10: Configurando o Driver - Serial.....	9
Figura 11: Configurando o Driver - Operação 1	9
Figura 12: Configurando o Driver - Operação 2	10
Figura 13: Configurando o Driver - Operação 3	10
Figura 14: Configurando o Driver - Operação 4	10
Figura 15: Configurando o Driver - Adicionando o LED	11
Figura 16: Configurando o Driver - Adicionando controle da camada física	12
Figura 17: Design da Tela Inicial.....	13

Figura 18: Adicionando botões	14
Figura 19: Adicionando associações nos botões	15
Figura 20: Configurando associação digital	16
Figura 21: Configurando associação por tabela.....	17
Figura 22: Configurando o tipo de coloração dos botões.....	18
Figura 23: Incluindo script no botão	19
Figura 24: Usando AppBrowser	20
Figura 25: Codificando o Liga e Desliga do LED.....	20
Figura 26: Executando a aplicação	21

1. PRÉ-REQUISITOS

1.1. Eclipse E3

Deve ser realizado o download do software Eclipse E3 e ter realizado a instalação do mesmo no computador através do [primeiro link](#) nas seções de referências deste arquivo.

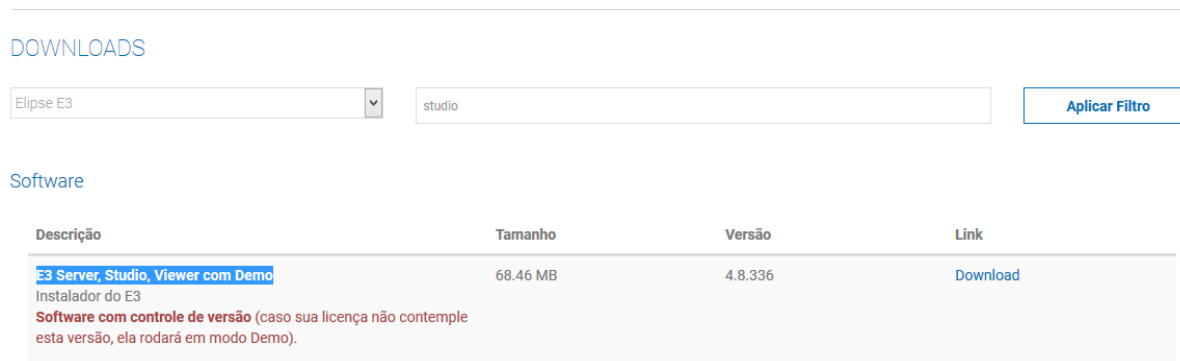


Figura 1: Download Software Eclipse E3

1.2. Driver de Comunicação

Realizar download do Driver Modicon Master (ASC/RTU/TCP) no site da Elipse através do [primeiro link](#) nas seções de referências deste arquivo.

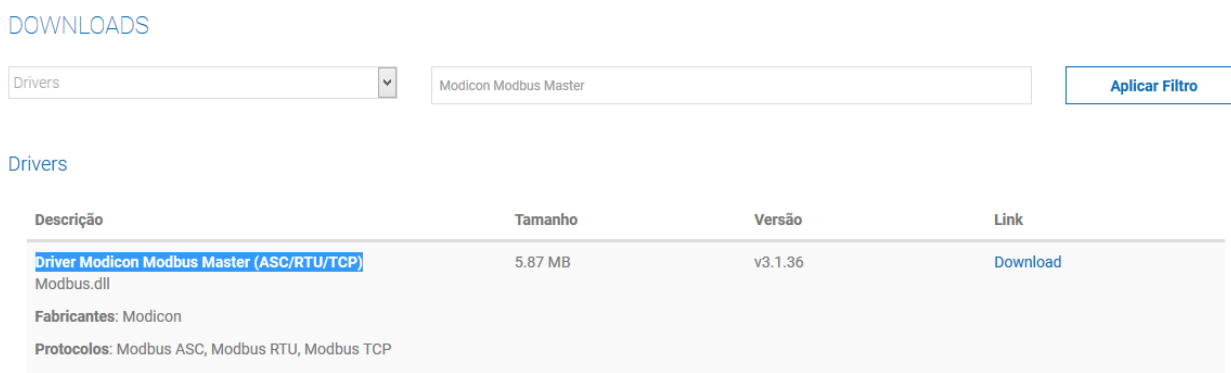


Figura 2: Download Driver Modbus

1.3. Arduino

Deve ser realizado o download do software Arduino e ter realizado a instalação do mesmo no computador através do [terceiro link](#) nas seções de referências deste arquivo.

Download the Arduino IDE



Figura 3: Downlaod Arduino IDE

1.4. Biblioteca ModbusSlave

Importe a biblioteca ModbusSlave para a IDE Arduino através do menu “**Sketch > Include Library > Add .ZIP Library**”, que pode ser adquirida no [segundo link](#) ou [quinto link](#) na seção de referência deste arquivo.

Após inclusão da biblioteca, abrir o arquivo “**ModbusSlave_0.ino**” dentro da pasta “**Arduino > ModbusSlave_0**” que pode ser adquirido no [segundo link](#) na seção de referência deste documento, como mostra a seguir:

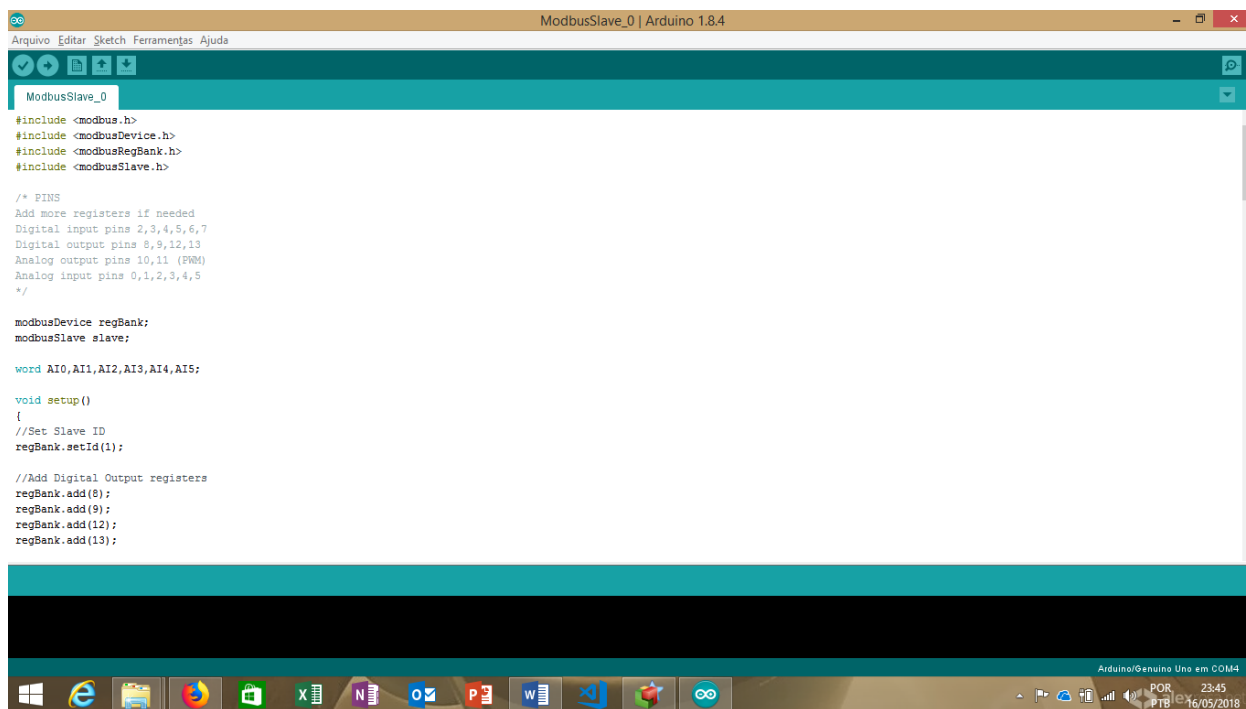


Figura 4: Simulando o Arduino como escravo Modbus

2. HARDWARE

Depois dos passos concluídos conforme mostra o tópico anterior, deve-se conectar um LED em uma protoboard ligados a saída digital 8 e ao GND do Arduino, como pode ser analisado no [quarto link](#) na seção de referências deste documento.

3. SOFTWARE

Após executar o programa “**studio.exe**”, deve-se criar um novo domínio, como segue:

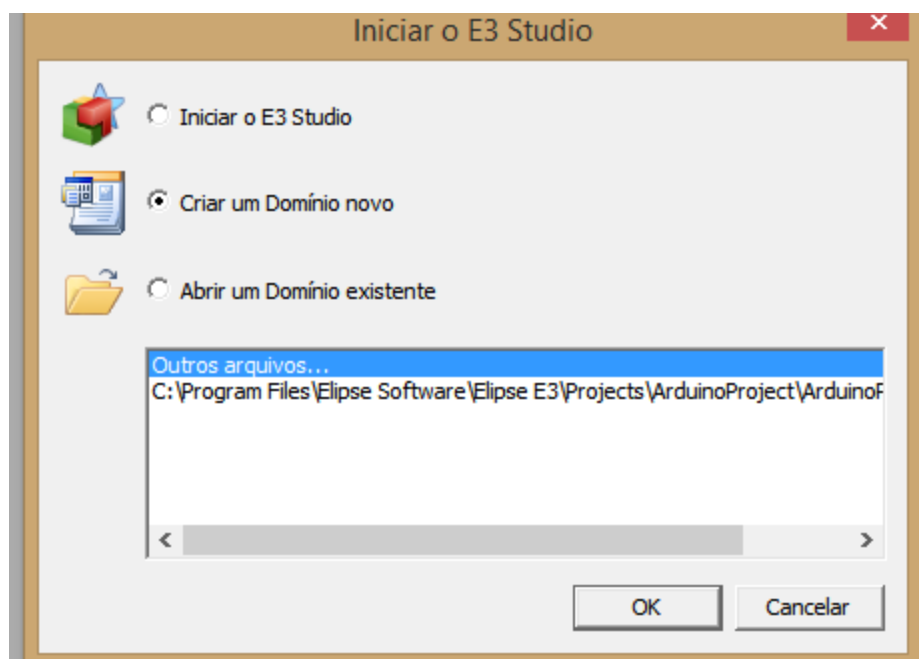


Figura 5: Criando um novo domínio

Seguir o assistente de aplicações clicando sempre em “Avançar”, lembrando aonde será gravada aplicação no campo “**Salvar a aplicação na pasta**”.

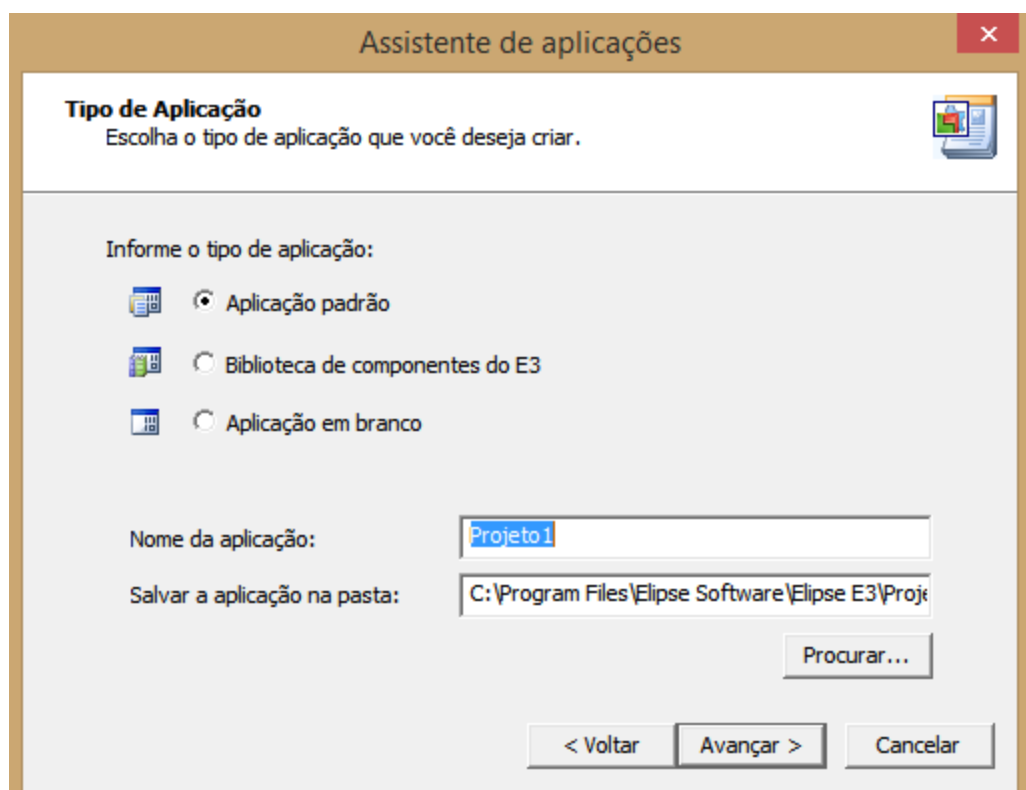


Figura 6: Configurando o Projeto

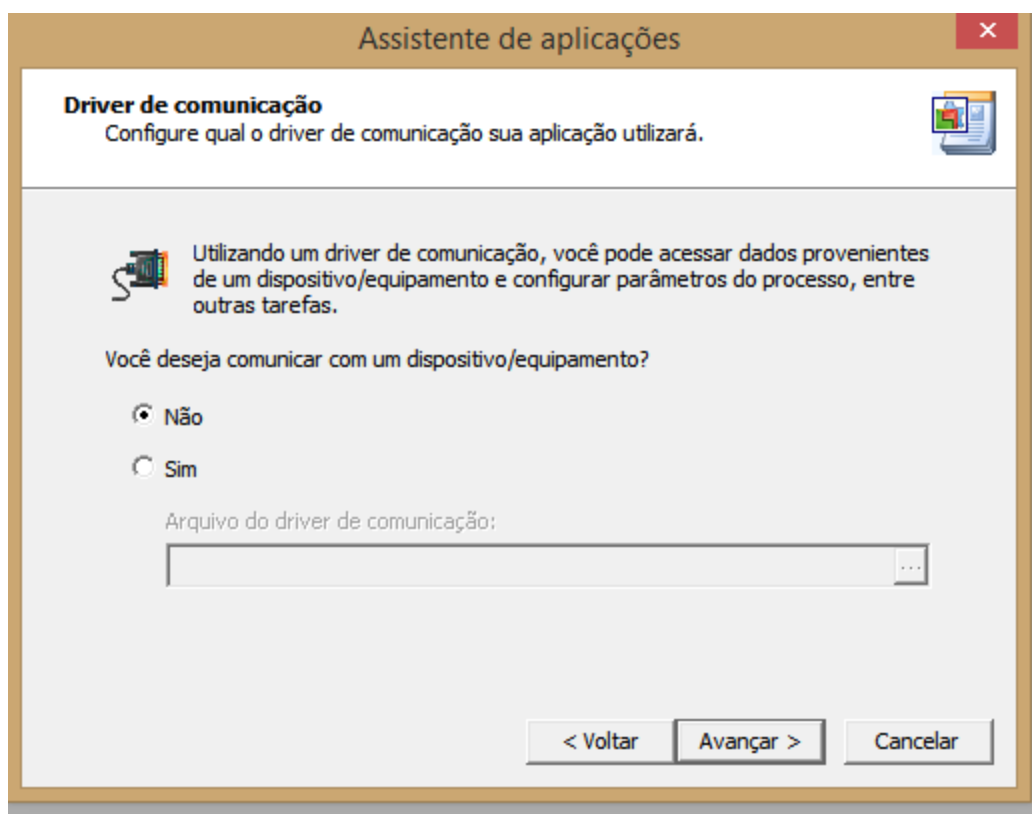


Figura 7: Seguindo o assistente de aplicações

Seguindo o assistente de aplicações corretamente, no final será exibido uma tela conforme segue:

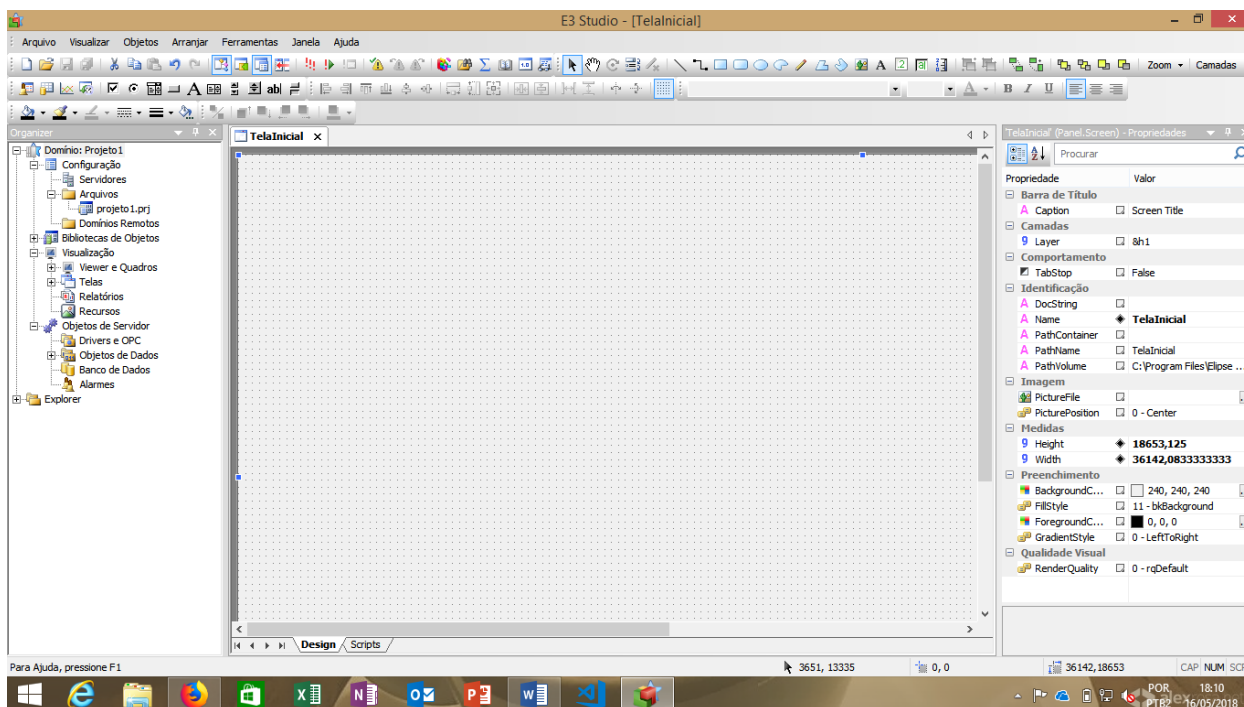


Figura 8: Aplicação configurada

Para melhor organização e manutenção do projeto, deve-se acessar o diretório em que foi salvo a propriedade, normalmente em **“C:\Program Files\Elipse Software\Elipse E3\Projects\Projeto1”** e criar uma arvore de pastas, como a seguir: **“Driver > Modbus”**. Dentro desta página deve ser colocada o driver de comunicação que foi adquirido na [seção 1.2](#).

No **“Organizer”** mostrado no canto esquerdo da aplicação, deve-se expandir a propriedade de **“Objetos de Servidor”** e clicar com o botão direito do mouse em **“Drivers e OPC”** e clicar em **“Inserir driver de comunicação em”** e selecionar o nome do projeto que foi escolhido no início do manual e selecionar o driver de comunicação.

Após esse procedimento, será aberto uma tela pedindo para realizar a configuração do driver: Na aba **Modbus** parametrize conforme figura a seguir:

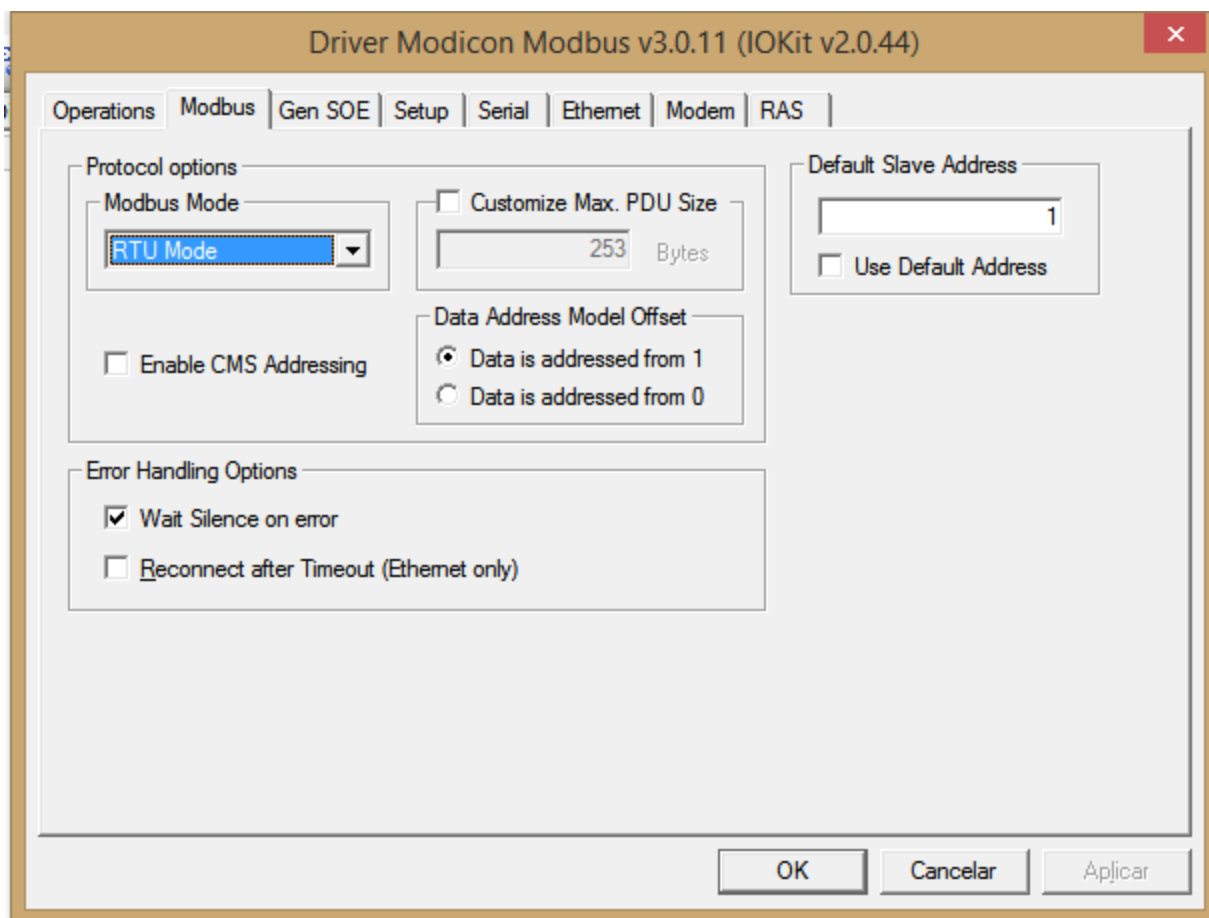


Figura 9: Configurando o Driver - Modbus

Na aba **Setup**, configure a propriedade *PhysicalLayer* como **serial** e na aba **Serial** conforme a imagem abaixo (vale lembrar que a “port” deve ser a mesma conectada pelo Arduino):

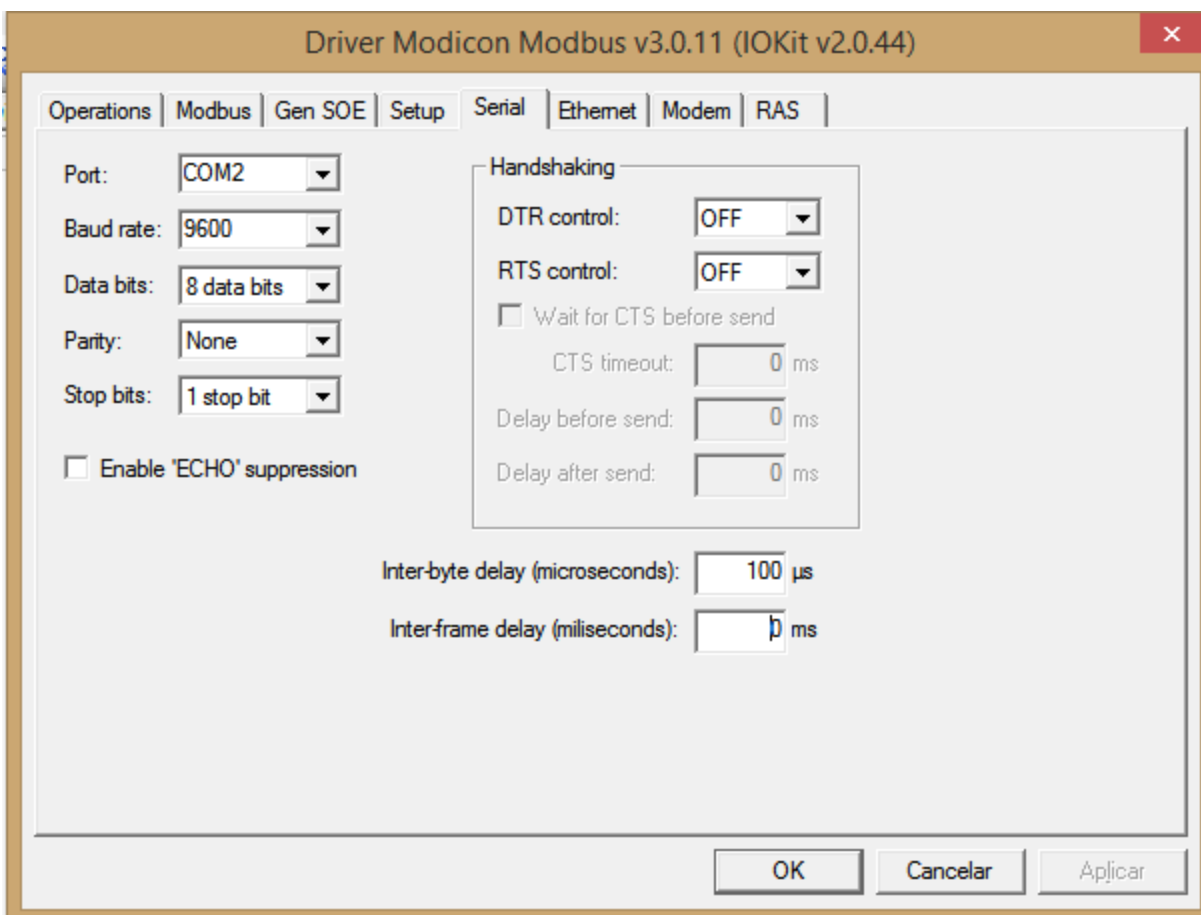


Figura 10: Configurando o Driver - Serial

Na aba de **Operations**, clique na operação em questão e clique em “**edit**”, depois configure as operações conforme imagens abaixo:

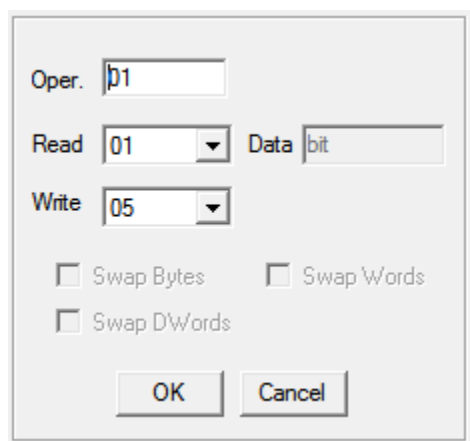


Figura 11: Configurando o Driver - Operação 1

Oper.

Read Data

Write

☐ Swap Bytes ☐ Swap Words

☐ Swap DWords

Figura 12: Configurando o Driver - Operação 2

Oper.

Read Data

Write Size

☐ Swap Bytes ☐ Swap Words

☐ Swap DWords ☐ Use Bit Mask

Figura 13: Configurando o Driver - Operação 3

Oper.

Read Data

Write Size

☐ Swap Bytes ☐ Swap Words

☐ Swap DWords ☐ Use Bit Mask

Figura 14: Configurando o Driver - Operação 4

Após essa configuração, clicar em **adicionar > tag de comunicação** e informar o nome do tag **“LED”**. Após isso, configurar o P1/N1, P2/N2, P3/N3 e P4/N4, conforme imagem a seguir:

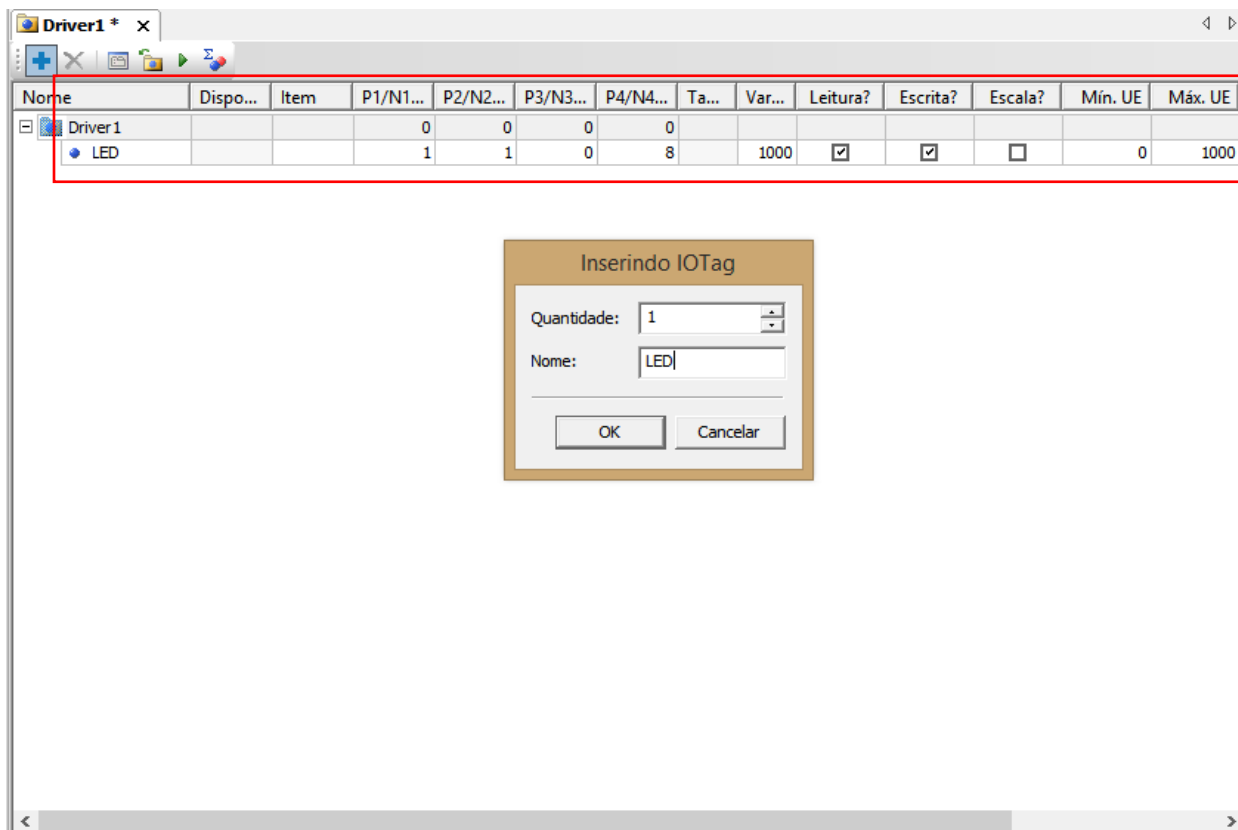


Figura 15: Configurando o Driver - Adicionando o LED

Clicar em **tag browser** e expandir a pasta **IOKit > General**, após isso arrastar o tag **IO.PhysicalLayerStatus** para o campo esquerdo, como na imagem a seguir:

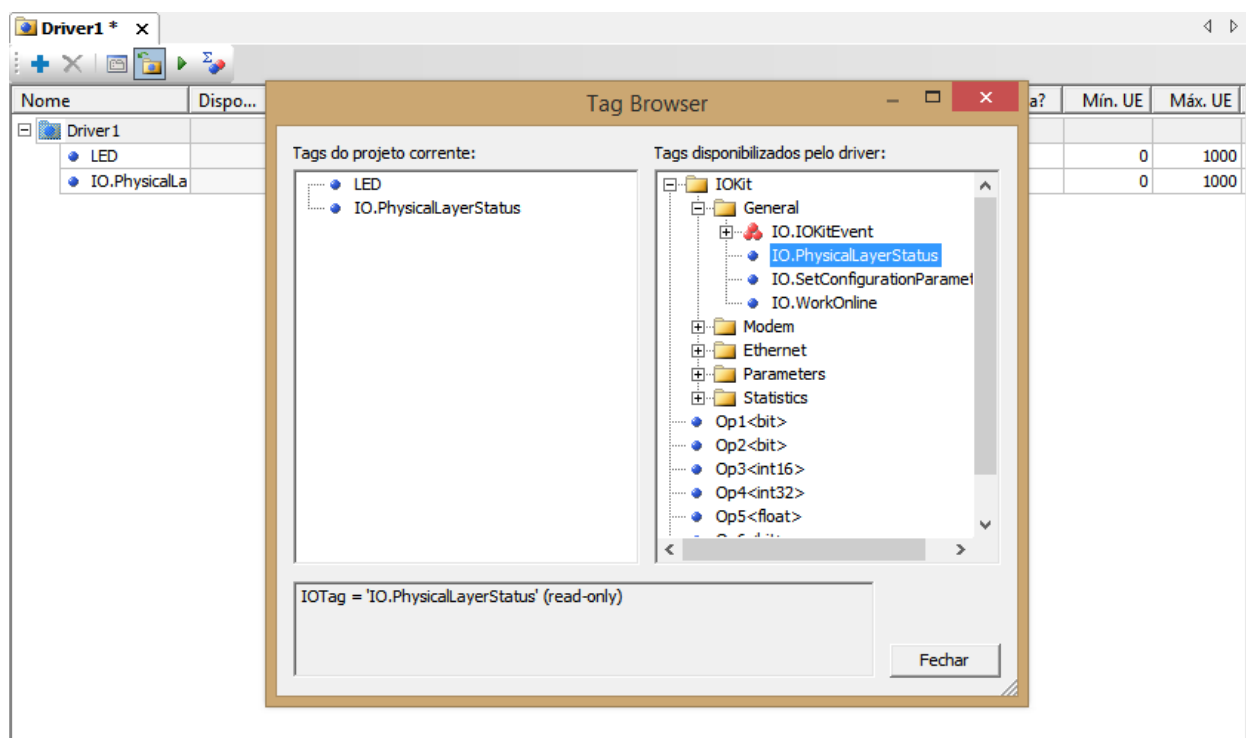


Figura 16: Configurando o Driver - Adicionando controle da camada física

No “**Organizer**”, expandir a opção de telas e dar dois cliques em **TelaInicial**, que foi criado automaticamente no início da aplicação. Verificar que esteja selecionado “**design**” na opção da tela, como na imagem a seguir:

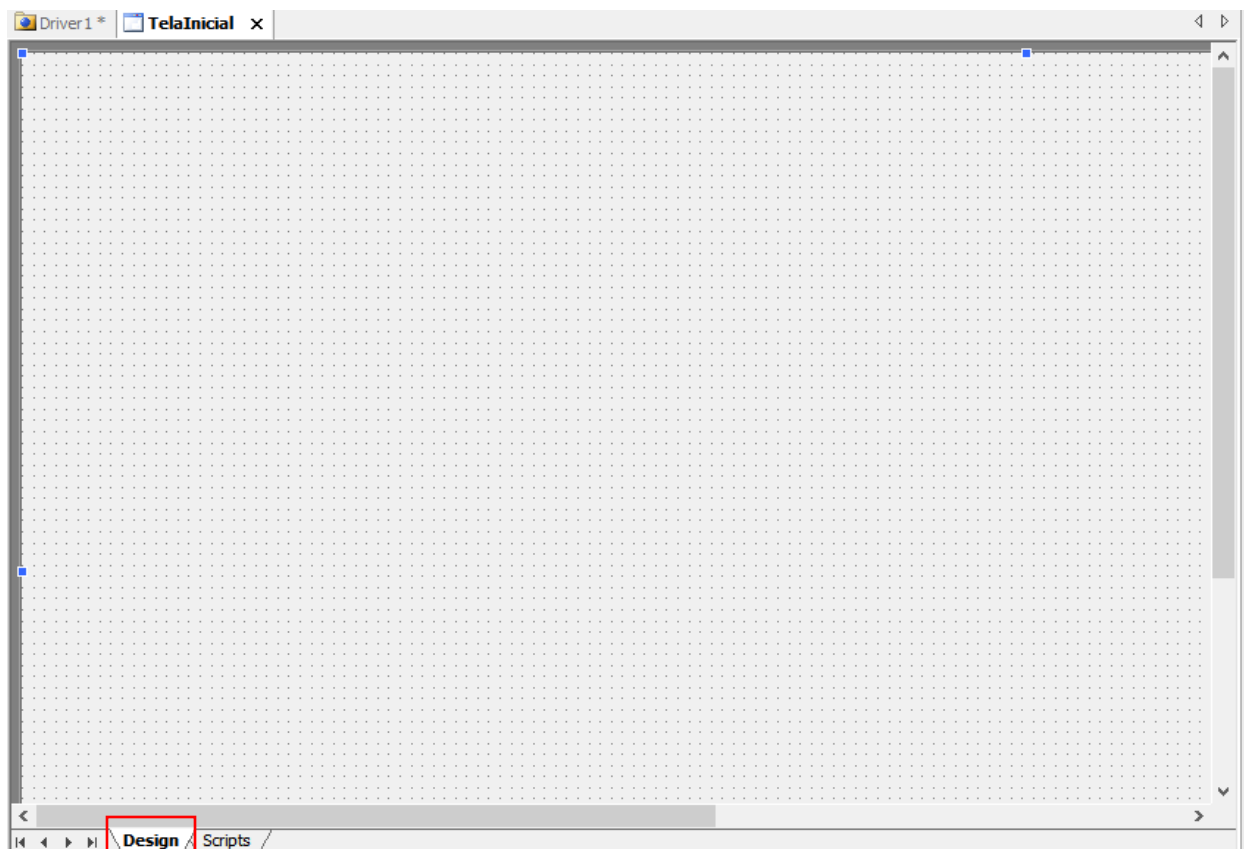


Figura 17: Design da Tela Inicial

Certifique-se que no menu de ferramentas no meu superior na opção “visualizar” esteja mostrando a opção “**Galeria**”. Com esta opção habilitada, clique em Galeria e arraste duas vezes o primeiro ícone da biblioteca de “**3-D Pushbuttons etc**” para o centro da **TelaInicial**, como segue na imagem:

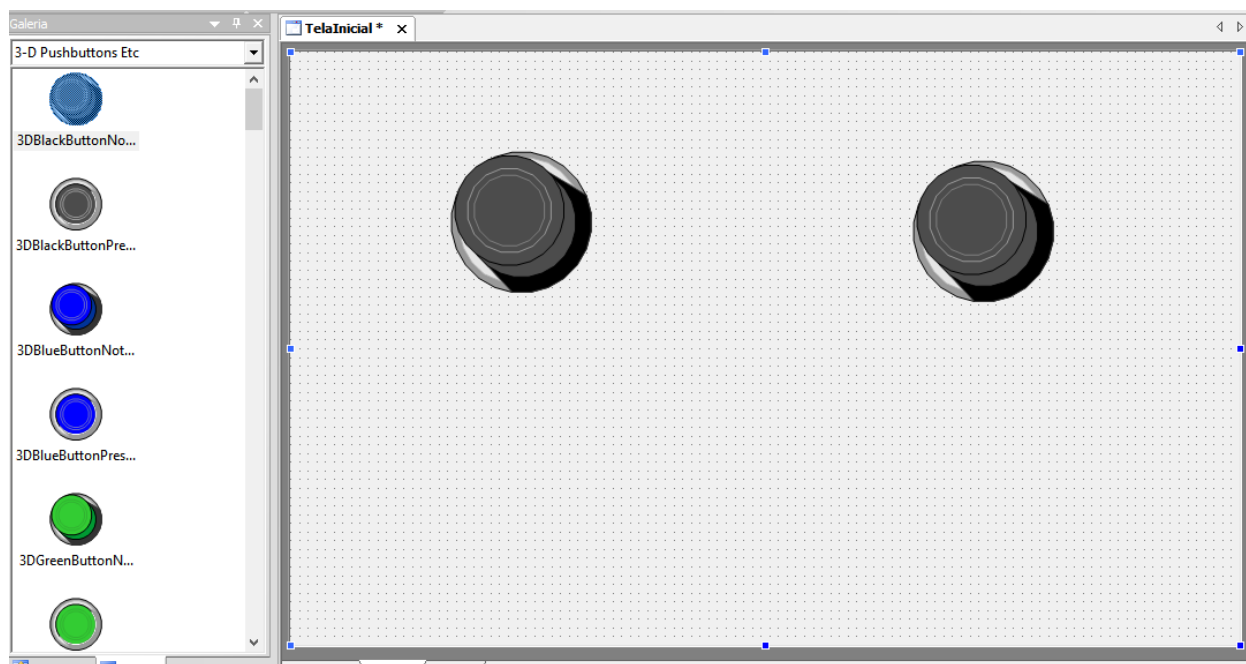


Figura 18: Adicionando botões

Clique com o botão direito do mouse no primeiro ícone, e clique em **“propriedades”** e selecione a aba de **“associações”**, conforme a imagem a seguir:

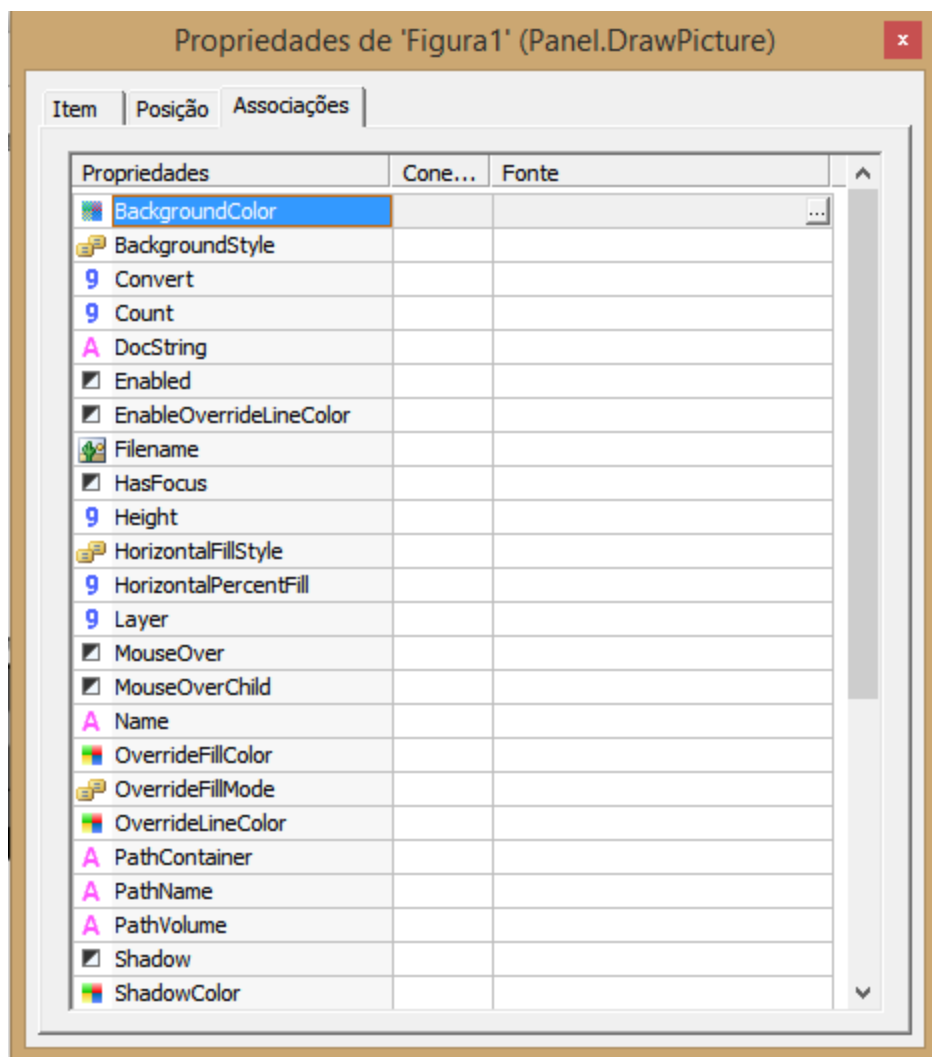


Figura 19: Adicionando associações nos botões

Crie uma **associação digital** na propriedade de **OverrideFillColor** com as cores que achar melhor e no campo **fonte** clique nos “3 pontinhos” e com o editor selecione **Servidor > Driver1 > Led > Value**, e clique em colar. Segue imagem:

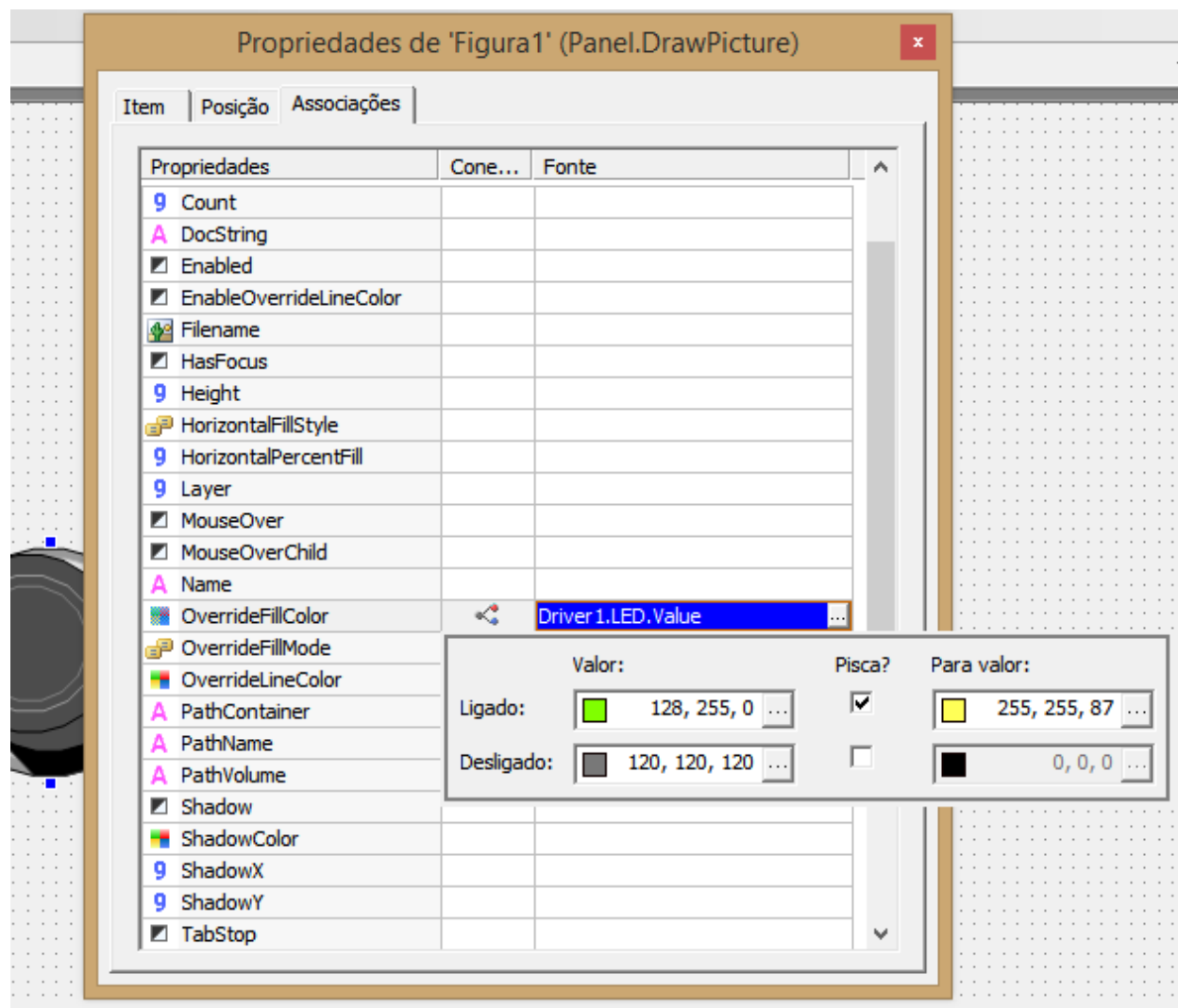


Figura 20: Configurando associação digital

Da mesma forma, realize o procedimento para o segundo botão, mas crie uma **associação por tabela** na mesma propriedade de **OverrideFillColor**, mas dessa vez, selecione **IO.PhysicalLayerStatus** dentro do driver, segue imagem com as opções:

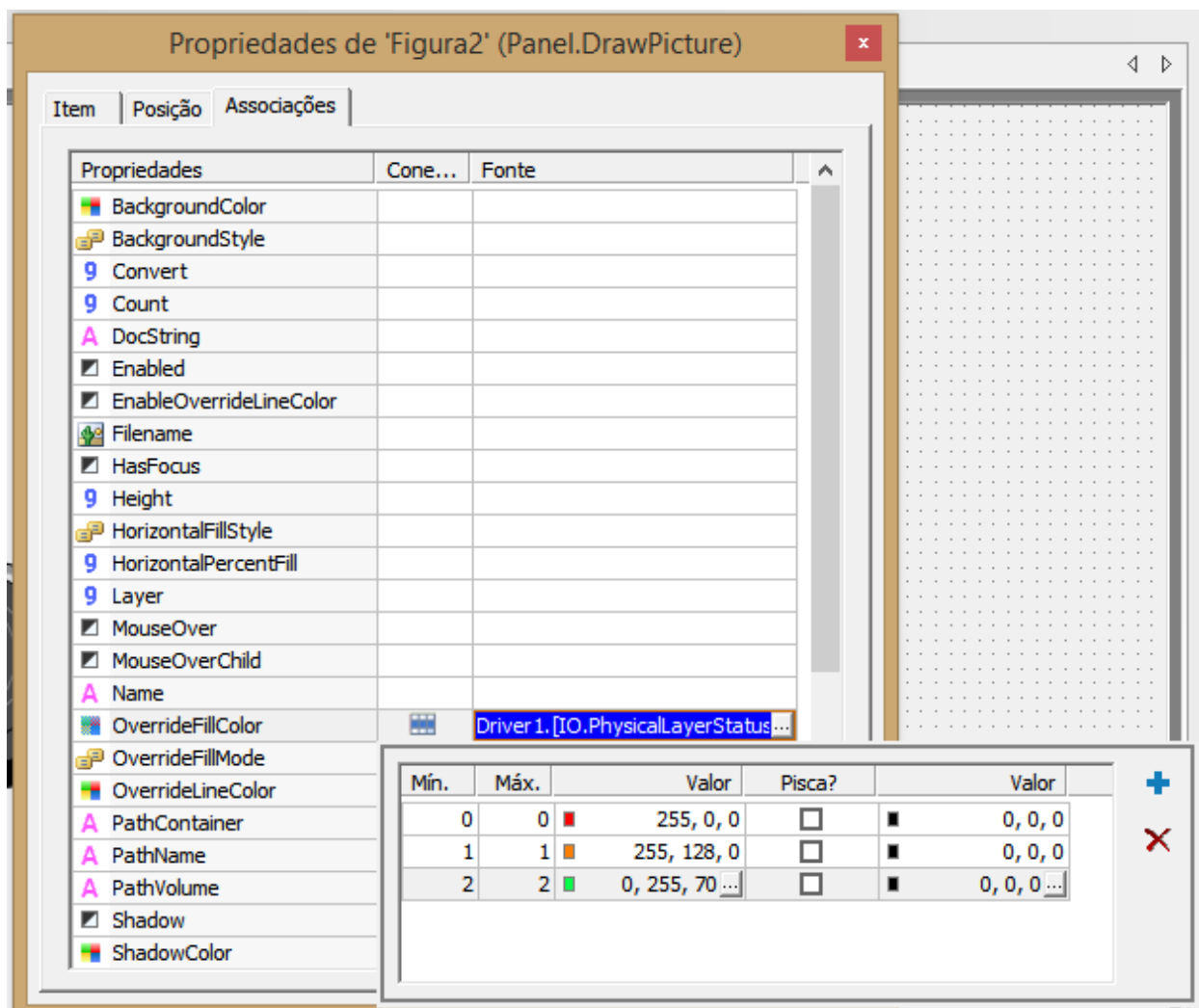


Figura 21: Configurando associação por tabela

Em ambos os botões, configure a propriedade de **OverrideFillMode** para “3 – ByBrightness”, conforme na imagem:

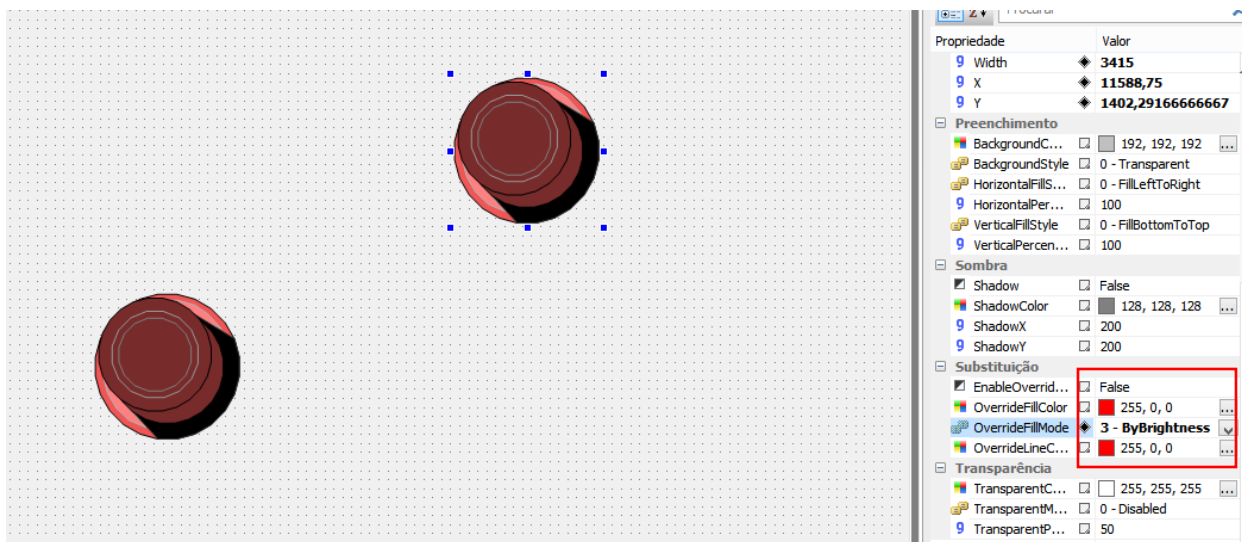


Figura 22: Configurando o tipo de coloração dos botões

No primeiro botão, onde fizemos uma associação digital, vamos então dar dois cliques sobre ele, assim será mostrado o editor de programação, clique na opção de **“script”**.

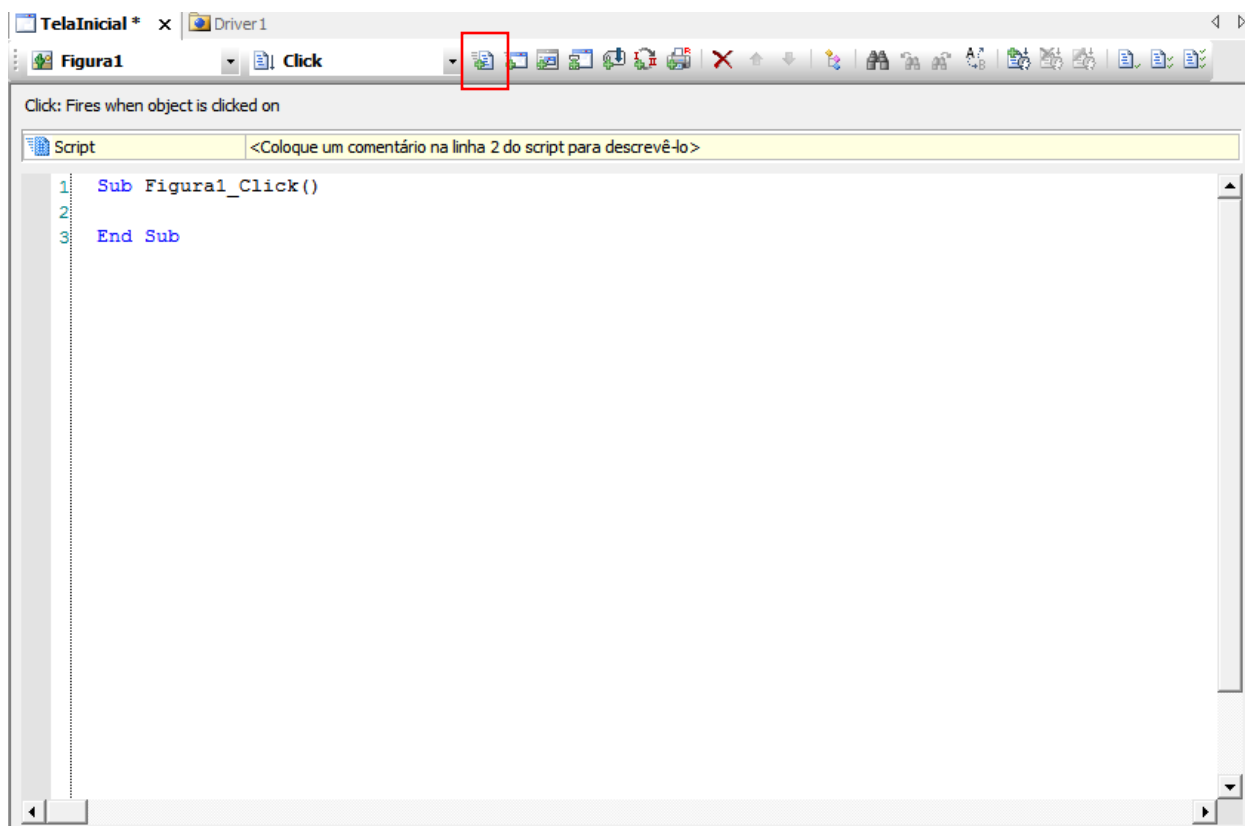


Figura 23: Incluindo script no botão

Precisamos saber qual a localização do LED que vamos acender, dentro da aplicação, e para isso vamos utilizar o **AppBrowser**, conforme segue:

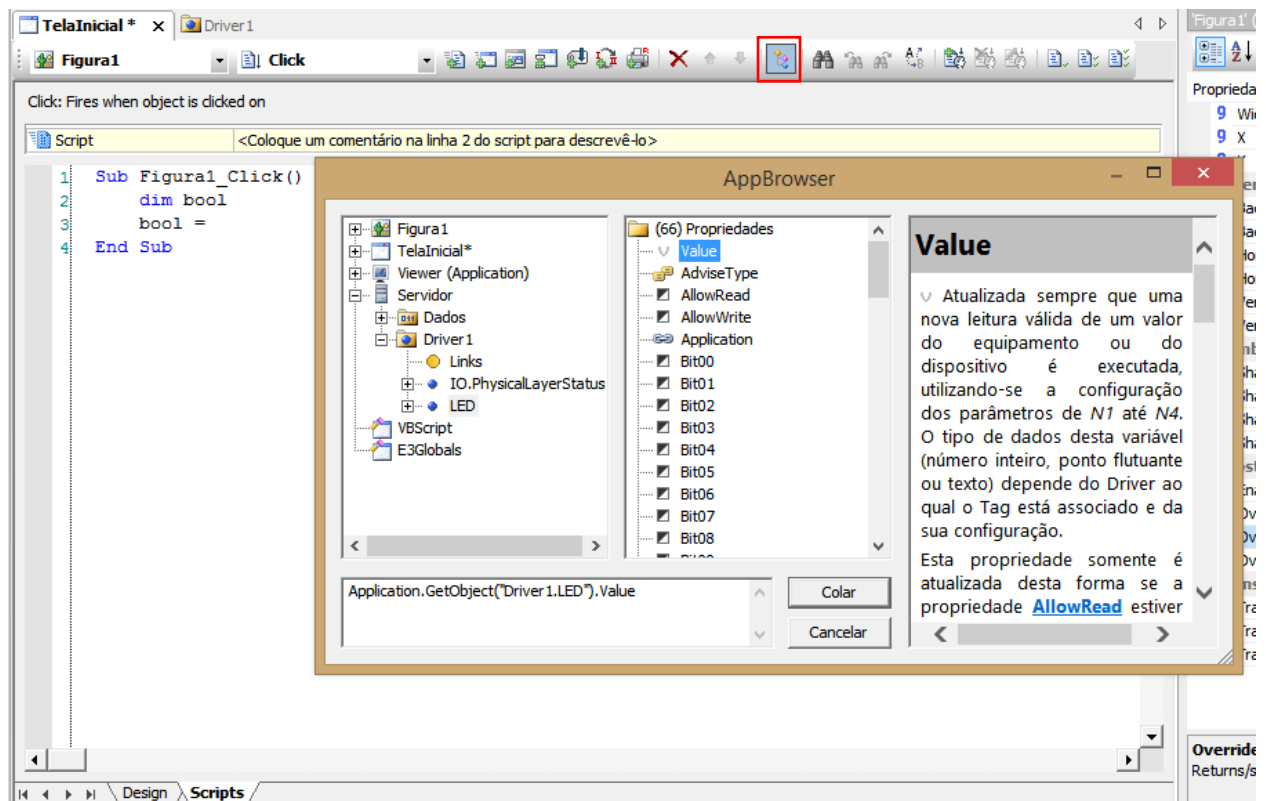


Figura 24: Usando AppBrowser

Por fim, o código deve se assemelhar com:

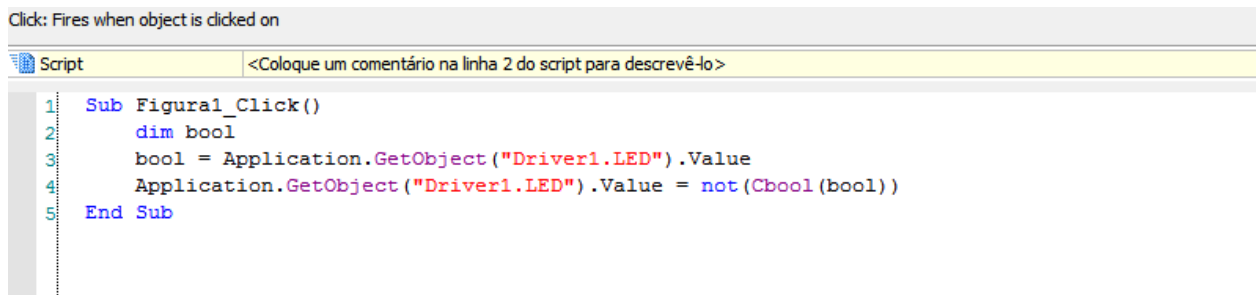


Figura 25: Codificando o Liga e Desliga do LED

Agora sim a aplicação está pronta para ser iniciada, clique em “Executar Aplicativo” para que a aplicação inicie:

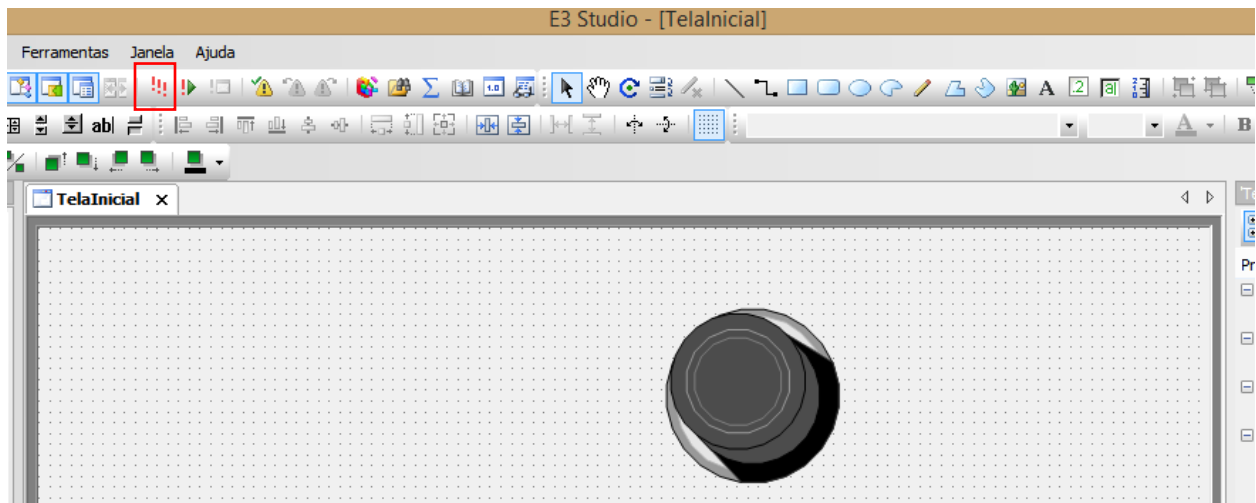


Figura 26: Executando a aplicação

REFERÊNCIAS

ELIPSE SOFTWARE. Disponível em <<https://www.elipse.com.br/downloads>>. Acesso em 16 de maio de 2018.

ELIPSE KNOWLEDGE. **Comunicando o Arduino com Elipse E3 e/ou Elipse SCADA**. Disponível em <<http://kb.elipse.com.br/pt-br/questions/5369/Comunicando+o+Arduino+com+Elipse+E3+e%7B47%7Dou+Elipse+SCADA>> Acesso em 16 de maio de 2018.

ARDUINO. Disponível em <<https://www.arduino.cc/en/Main/Software>>. Acesso em 16 de maio de 2018.

FATECINO. **Projeto No. 1 Controle de um LED**. Disponível em <http://fatecjd.edu.br/fatecino/arg_projetos/02-Projeto-1-LED.pdf>. Acesso em 16 de maio de 2018.

GOOGLE. **arduino-modbus-slave**. Disponível em <<https://code.google.com/archive/p/arduino-modbus-slave/downloads>> Acesso em 16 de maio de 2018.