

GILIARDI OLIVEIRA

CONSTRUTOR DE INTERFACE DE SUPERVISÃO E CONTROLE
Manual do Usuário

São Leopoldo
2018

LISTA DE FIGURAS

Figura 1 – Tela inicial do Programa de Instalação Scada IDE	5
Figura 2 - Finalizando o assistente de Instalação do Scada IDE	5
Figura 3 – Conjunto de atalhos na área de trabalho	6
Figura 4 – diretório dos arquivos	6
Figura 5 – Caixa de confirmação de desinstalação do software	7
Figura 6 – Diagrama de blocos do Scada IDE	7
Figura 7 - Menu do programa Scada Design.....	8
Figura 8 - Janela: Importar Configurações do Arquivo	10
Figura 9 - Janela de Detalhes do Componente	11
Figura 10 - Caixa de Seleção de Componente.....	12
Figura 11 - Coordenadas top e left do componente Botão	12
Figura 12 - Acesso a janela de Configurações de Entrada	14
Figura 13 - janela de Configurações dos Canais de Entrada	15
Figura 14 - Janela de Configuração do Projeto SCADA.....	16
Figura 15 – Tela da aplicação Manager	17
Figura 16 – Janela de seleção de projeto.....	18
Figura 17 – Campo de definição da versão do firmware	23
Figura 18 – Janela de Configurações do Projeto	23
Figura 19 – diagrama do projeto exemplo	25

SUMÁRIO

1 INTRODUÇÃO	4
2 GERENCIAMENTO DO SISTEMA.....	4
2.1 Instalação.....	4
2.2 Desinstalação.....	6
3 DESCRIÇÃO DOS SISTEMAS	7
3.1 Sistema SCADA DESIGN	8
3.1.1 Importar endereços do hardware.....	9
3.1.2 Edição das Propriedades do Componente	11
3.2 Sistema SCADA CONTROL.....	15
3.3 Sistema MANAGER	17
3.4 Sistema RegisterExt.....	18
4 CARACTERÍSTICAS DO FIRMWARE	19
4.1 Programação do Firmware.....	19
4.2 Pinos de entrada ou saída	19
4.3 Entradas Analógicas	20
4.4 Interpretador de comandos do micro controlador	21
4.5 Envio de dados da serial para o Scada Control	24
5 PROJETO EXEMPLO	25

1 INTRODUÇÃO

Este manual tem como objetivo orientar o usuário durante a instalação do pacote de software Scada IDE. Para um melhor entendimento, o conteúdo foi simplificado, deixando apenas as partes necessárias à instalação e ao entendimento do sistema.

2 GERENCIAMENTO DO SISTEMA

O pacote de software Scada IDE está disponível apenas para sistemas operacionais Windows. Para o funcionamento do sistema, não há necessidade de instalação de pacotes adicionais.

O sistema foi desenvolvido em linguagem *Object Pascal*, Delphi 10.1 Berlin na versão 24.0.22858.6822. A tabela 1 exibe a versão de cada uma das aplicações que compõem o pacote Scada IDE.

Tabela 1 – Versão dos Sistemas

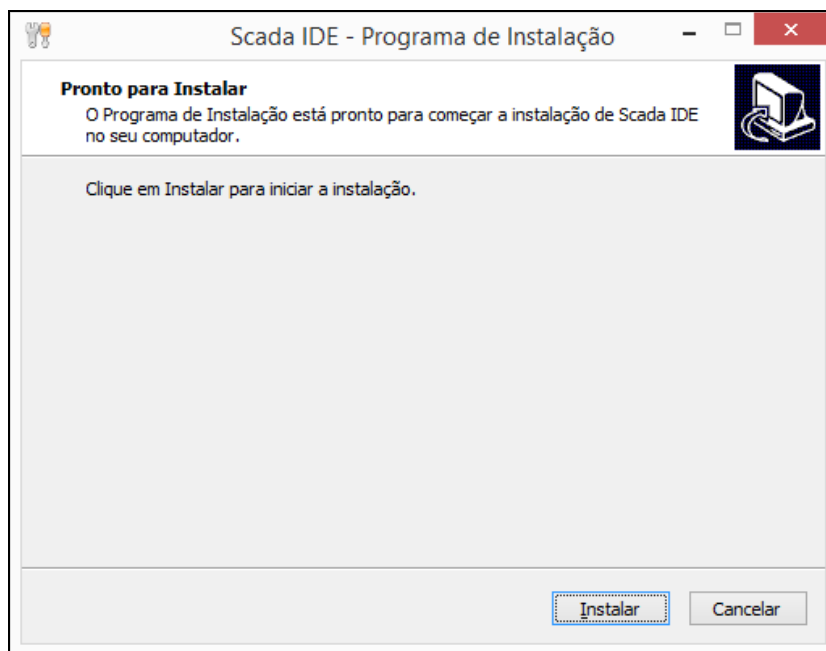
Sistema	Versão
Scada Design	1.0.0.7
Scada Control	1.0.0.8
Manager	1.0
RegisterExt	1.0

O arquivo de instalação encontra-se disponível para download no endereço <https://github.com/giliardi81/scadaIDE>

2.1 Instalação

Para instalação do pacote Scada IDE, execute o arquivo de instalação “setup.exe”. A tela inicial de instalação será exibida, conforme figura 1.

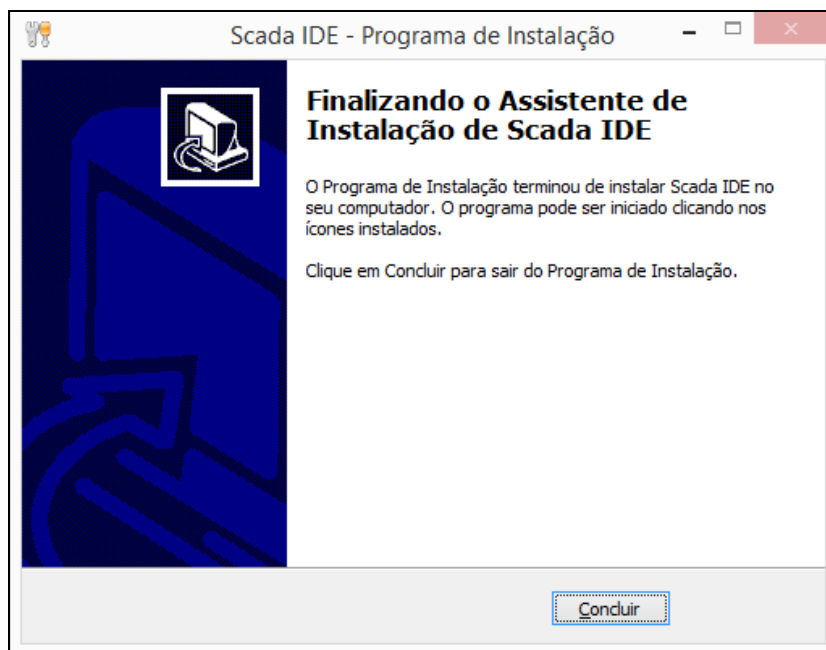
Figura 1 – Tela inicial do Programa de Instalação Scada IDE



Fonte: Elaborado pelo autor.

Selecionando a opção “Instalar”, o processo de instalação ocorrerá de forma automática. Após a instalação a tela “Finalizando o assistente de Instalação do Scada IDE” será exibida, como demonstra a figura 2.

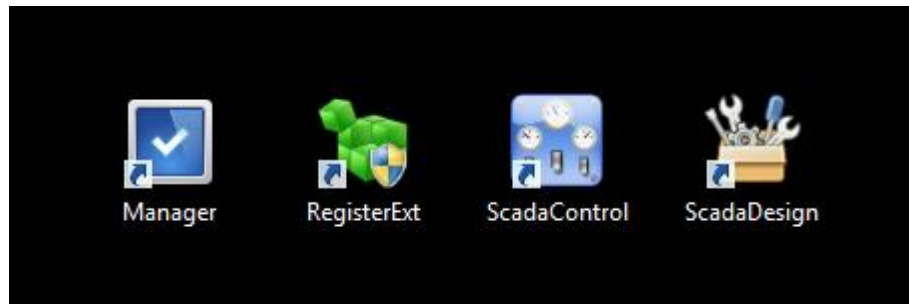
Figura 2 - Finalizando o assistente de Instalação do Scada IDE



Fonte: Elaborado pelo autor.

Ao final do processo de instalação, será criado um conjunto de atalhos na área de trabalho. A figura 3 exibe os atalhos dos programas.







Figura 3 – Conjunto de atalhos na área de trabalho



Fonte: Elaborado pelo autor.

O diretório padrão da instalação do sistema é “C:\Program Files (x86)\Scada”. Conforme demonstra a figura 4.

Figura 4 – diretório dos arquivos

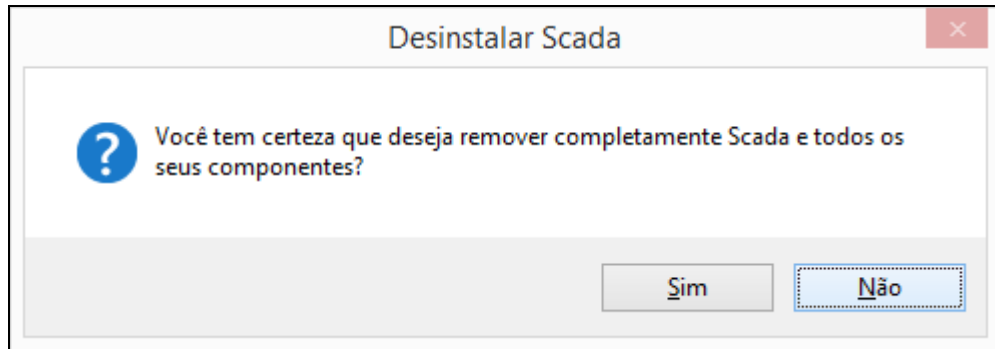
« Disco Local (C:) » Arquivos de Programas (x86) » Scada			
Nome	Data de modificaç...	Tipo	
 Manager	28/05/2018 20:34	Aplicativo	
 RegisterExt	28/05/2018 20:47	Aplicativo	
 ScadaControl	29/05/2018 12:33	Aplicativo	
 ScadaDesign	29/05/2018 13:46	Aplicativo	
 unins000.dat	31/05/2018 10:03	Arquivo DAT	
 unins000	31/05/2018 10:03	Aplicativo	

Fonte: Elaborado pelo autor.

2.2 Desinstalação

Para executar o processo de desinstalação do software basta executar o arquivo “unins000” que se encontra no diretório da instalação. Uma caixa de diálogo será aberta solicitando a confirmação do usuário. A figura 5 exibe a caixa de confirmação do processo de desinstalação.

Figura 5 – Caixa de confirmação de desinstalação do software

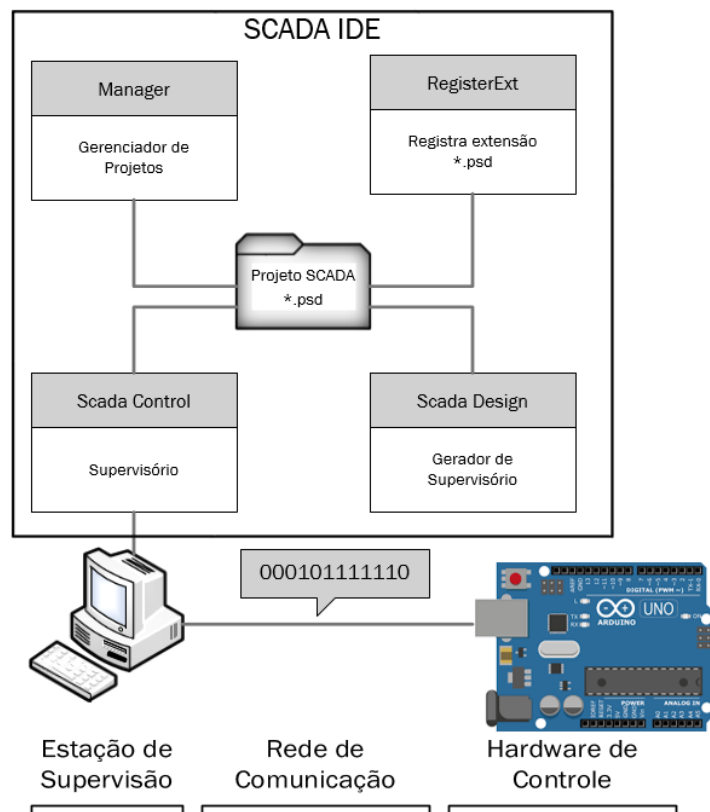


Fonte: Elaborado pelo autor.

3 DESCRIÇÃO DOS SISTEMAS

Abaixo uma breve descrição do conjunto de sistemas que compõe o Scada IDE. A figura 6 demonstra o diagrama de blocos do sistema.

Figura 6 – Diagrama de blocos do Scada IDE



Fonte: Elaborado pelo autor.

3.1 Sistema SCADA DESIGN

O software Scada Design permite a criação de projetos, onde é permitido inserir, remover e alterar as propriedades dos componentes.

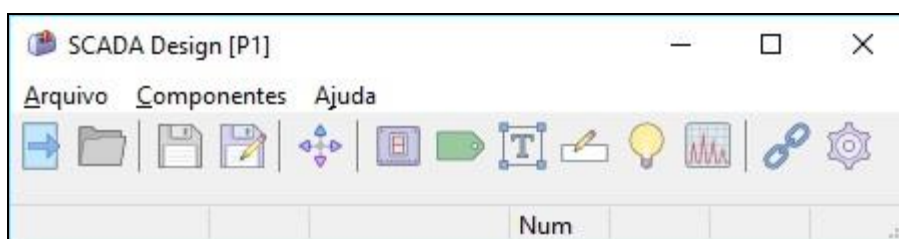
Ao iniciar a aplicação, podemos criar um novo projeto ou abrir um projeto existente. Ao criarmos um novo projeto, por padrão este ficará salvo dentro da pasta “Projetos” que se encontra junto ao diretório da aplicação. Uma vez criado o projeto, também temos a opção de “Salvar Projeto Como...”, onde é permitido definir outro diretório para salvar o projeto. O arquivo de projeto tem sua extensão *.psd (Projeto Scada Design).

Ao projeto, podemos inserir ou remover componentes e definir sua posição dentro da área interna do form. Também é possível inserir ou remover a imagem que vai compor o layout do projeto. Os componentes dentro do projeto vão apresentar três comportamentos distintos, sendo:

- Escrita: quando enviam mensagens para o hardware externo via comunicação serial;
- Leitura: quando indicam status ou valores de variáveis recebidas do hardware externo via comunicação serial.
- Estático: não executam ação, são utilizados apenas para compor o layout da tela. São o componente imagem e etiqueta.

Para a interligação com o driver de comunicação foi adicionado a cada componente uma propriedade chamada link, que vai corresponder ao endereço da variável dentro do firmware. Desta forma temos um identificador único dentro do sistema que permite a comunicação serial entre o componente em tela com o hardware. A figura 7 representa a janela principal do programa Scada Design.

Figura 7 - Menu do programa Scada Design



Fonte: Elaborado pelo autor.

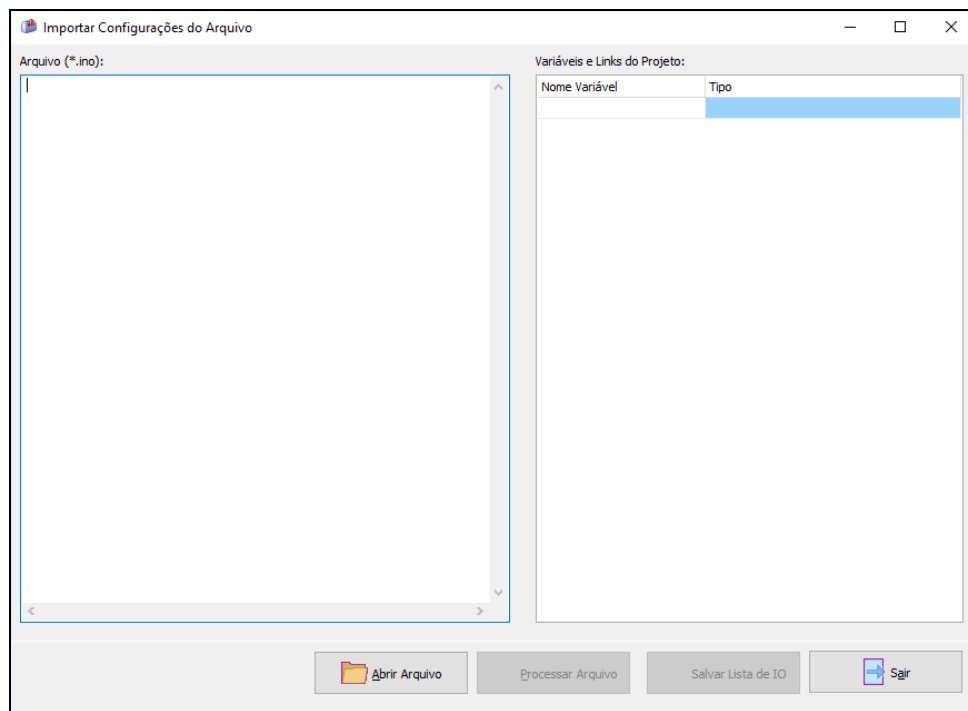
A respectiva função dos ícones da direita para esquerda é:

- Encerrar Aplicação;
- Abrir um projeto existente;
- Salvar Design;
- Salvar Projeto como;
- Mover Componente;
- Inserir Botão;
- Inserir Etiqueta;
- Inserir Bloco de Leitura;
- Inserir Bloco de Escrita;
- Inserir Bloco de Indicação;
- Inserir Gráfico;
- Importar Configurações do Arquivo;
- Configurações do Projeto;

3.1.1 Importar endereços do hardware

Para tornar o processo de ligação entre a propriedade link de um componente dentro do Scada Design com o firmware, foi adicionado um recurso para importar os nomes dos endereços de variáveis, entradas e saídas. Dentro do ambiente basta acessar o botão “Importar informações do arquivo”. A figura 8 representa a janela de Importar Configurações do Arquivo.

Figura 8 - Janela: Importar Configurações do Arquivo



Fonte: Elaborado pelo autor.

Ao clicar sobre o botão “Abrir Arquivo” selecione o arquivo que deseja importar as configurações. Após abrir o arquivo, o botão “Processar Arquivo” é liberado. Clique sobre o botão “Processar Arquivo” e a lista de variáveis e links será gerada. Ao final o botão “Salvar Lista de IO” é liberado. Ao clicar em “Salvar Lista de IO” quatro arquivos são salvos dentro da pasta do projeto. Os arquivos gerados são:

analogList: contém a lista de variáveis do tipo analógico. São identificadas pela chave “analogRead”;

inputList: contém a lista de variáveis correspondente as entradas. São identificadas pela chave “pinMode” seguidas do identificador “INPUT_PULLUP”.

outputList: contém a lista de variáveis correspondentes as saídas. São identificadas pela chave “pinMode” seguidas do identificador “OUTPUT”.

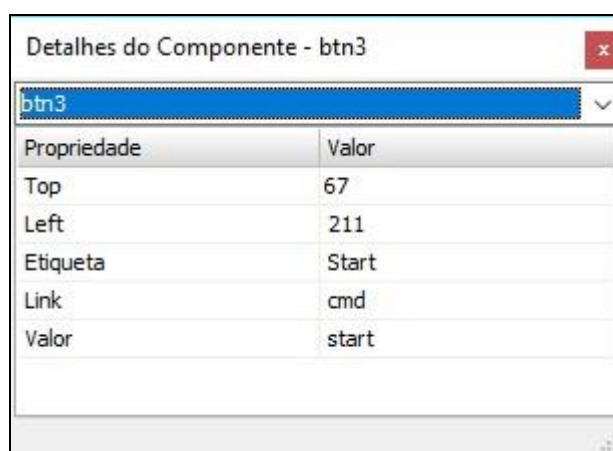
linkList: contém a lista de variáveis serão manipuladas pela serial (escrita e leitura). São identificados pelos identificadores “beginVAR” e “endVAR”;

statusList: contém a lista de variáveis do tipo booleana. São utilizadas pelo componente Led para indicar status na tela do supervisor. São identificados pelos identificadores “beginSTATUS” e “endSTATUS”;

3.1.2 Edição das Propriedades do Componente

Para a edição das propriedades de um componente basta clicar sobre ele que a janela de “Detalhes do Componente” é exibida, conforme visualizamos na figura 9.

Figura 9 - Janela de Detalhes do Componente

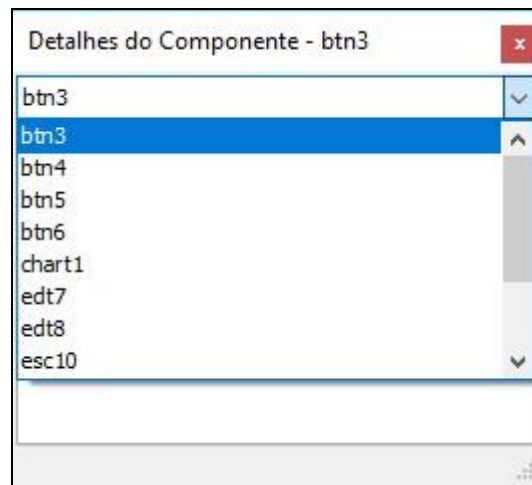


Fonte: Elaborado pelo autor.

Quando ocorre a seleção de um componente, suas propriedades são carregadas na tabela abaixo da caixa de seleção. A coluna a direita identifica o nome da propriedade, e a coluna ao lado o seu valor. Ao alterar um valor é necessário confirmar a alteração clicando em “Enter”.

Também é possível selecionar um componente dentro da caixa de seleção na parte superior do form. Todos os componentes do projeto são exibidos na lista e são identificados pela propriedade “name”. A figura 10 exibe a caixa de seleção de componentes, com a seleção do componente btn3.

Figura 10 - Caixa de Seleção de Componente



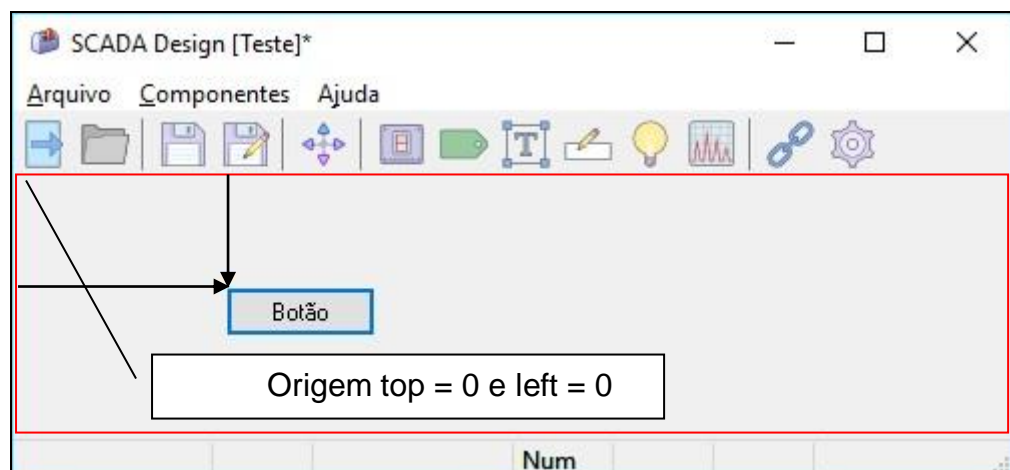
Fonte: Elaborado pelo autor.

As propriedades são divididas em dois tipos:

- Propriedades comuns: são propriedades que se repetem em todos os componentes visuais como a sua posição em relação a janela. São respectivamente a propriedade *top* e *left*.
- Propriedades específicas: são propriedades únicas que cada componente possui.

As propriedades *top* e *left* são determinadas em relação as dimensões da área interna da janela principal que vai armazenar a imagem que irá representar o sinótico do projeto. A figura 11 exemplifica como são determinadas as posições. O contorno em vermelho determina a área disponível para a imagem.



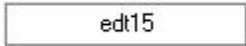




Figura 11 - Coordenadas top e left do componente Botão



Fonte: Elaborado pelo autor.

A tabela 2 detalha a imagem do componente na tela, seu ícone da barra de menu, suas propriedades e uma descrição breve da mesma.

Tabela 2 – Propriedades específicas dos componentes

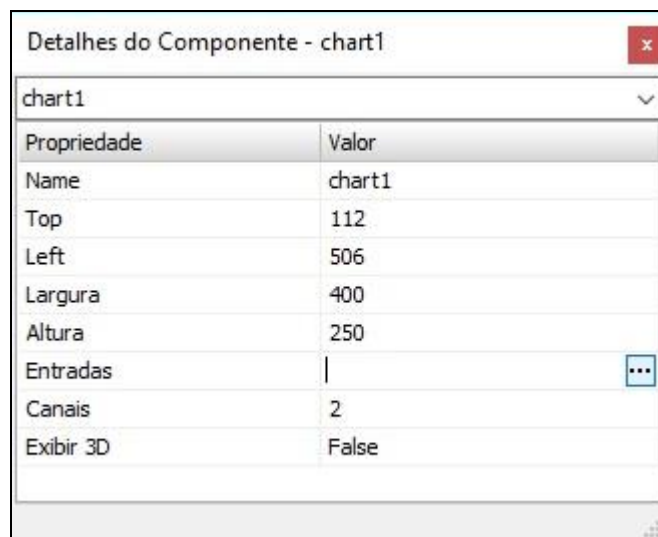
Componente	Imagem do Componente em Tela	Ícone da barra de menu	Propriedades	Descrição
Botão			Etiqueta	Rótulo do botão
			Link	Nome da variável dentro do firmware
			Valor	Valor a ser enviado quando executar o evento OnClick
Etiqueta			Auto Ajustar	Ajusta o seu tamanho ao conteúdo da etiqueta
			Etiqueta	Texto da etiqueta
			Tamanho da Fonte	Tamanho da fonte do bloco de leitura
Bloco Leitura			Cor da Fonte	Cor da fonte do bloco de leitura
			Cor do Fundo	Cor do fundo do bloco de leitura
			Link	Nome da variável dentro do firmware
Bloco Escrita			Somente Números	Permite somente caracteres numéricos
			Busca valor do HD	Ao conectar ao hardware atualiza o valor corrente da variável
			Link	Nome da variável dentro do firmware
Bloco de Indicação			Modelo	Tipo de indicador. Pode ser redondo ou retangular
			Cor	Define a cor do indicador. Pode ser verde, amarelo, vermelho ou azul
			Link	Nome da variável dentro do firmware
Gráfico			Largura	Largura do componente gráfico
			Altura	Altura do componente gráfico
			Entradas	Abre a janela de

	Configurações dos Canais de Entrada
Canais	Número de Canais de Entrada
Exibir 3D	Exibe o gráfico em 3D

Fonte: Elaborado pelo autor.

O componente Gráfico é composto por mais uma janela de configuração. A janela de configuração do gráfico permite gerenciar os canais de entrada. Para acessar a janela de configuração dos canais de entrada selecione o botão ao lado da propriedade “Entradas”, conforme detalhado na figura 12.

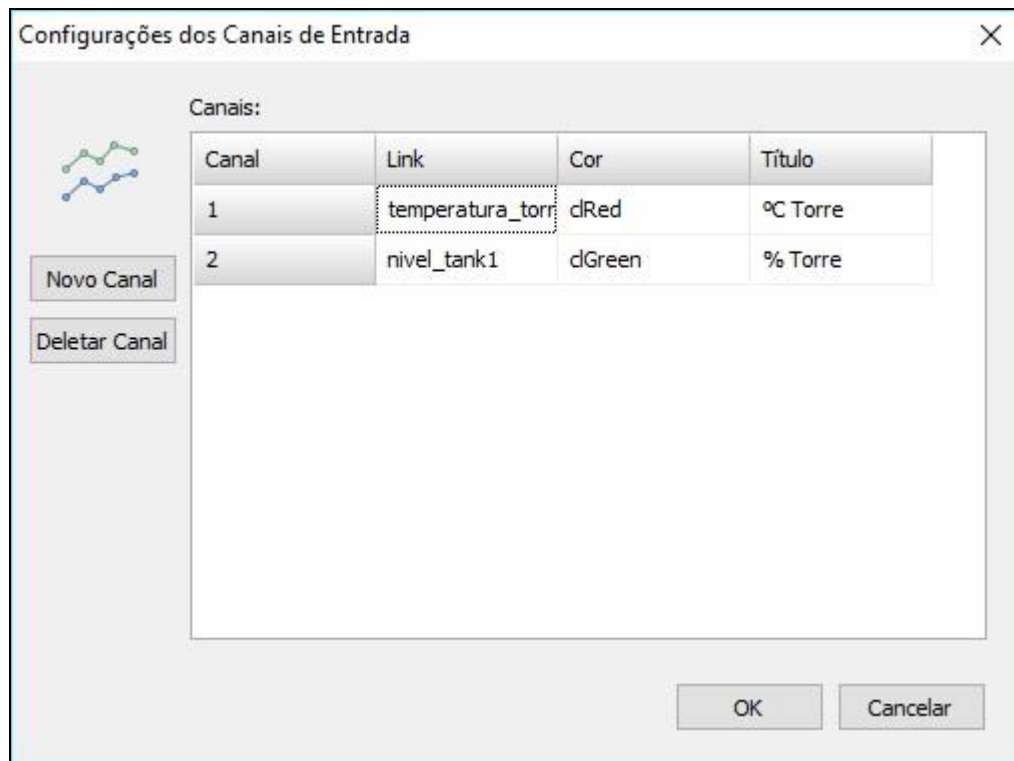
Figura 12 - Acesso a janela de Configurações de Entrada



Fonte: Elaborado pelo autor.

Ao clicar sobre o botão, a janela de Configurações dos Canais de Entrada é exibida, conforme figura 13.

Figura 13 - janela de Configurações dos Canais de Entrada



Fonte: Elaborado pelo autor.

Ao gráfico é permitido inserir até cinco canais de entrada. Cada canal possui três propriedades:

- Link: endereço do firmware;
- Cor: cor da linha do gráfico;
- Título: título do respectivo gráfico;

Quando o projeto se encontra em execução na aplicação Scada Control, os canais são plotados dentro da região do gráfico a cada intervalo de tempo definido no parâmetro “tempoDelay” do arquivo do micro controlador.

3.2 Sistema SCADA CONTROL

O sistema Scada Control é responsável por realizar a interface da aplicação em execução dentro do firmware com o arquivo de projeto definido. Ao executar a aplicação, uma comunicação bidirecional entre supervisor e firmware é estabelecida.

A arquitetura utilizada torna o microcontrolador o responsável por executar a lógica de controle. Ao supervisor, é permitido a escrita aos endereços definidos na

configuração do projeto, bem como a leitura. A falta de comunicação do supervisor não compromete o funcionamento do microcontrolador.

Ao inicializar a aplicação o sistema verifica a existência do arquivo de configuração SCconfig.ini. Este arquivo contém uma chave que aponta para o diretório dos arquivos do projeto Scada Design. Se o arquivo é encontrado, o projeto definido no arquivo é carregado. Caso o arquivo não for encontrado a janela de Configuração do Projeto é exibida, solicitando ao usuário definir um arquivo de projeto Scada Design. Se o usuário cancelar a operação, a aplicação é finalizada. A figura 14 exibe a tela de Configuração do Projeto.

Figura 14 - Janela de Configuração do Projeto SCADA



Fonte: Elaborado pelo autor.

Ao carregar a aplicação, todos os componentes e as respectivas configurações realizadas no projeto Scada Design são carregadas. Para inicializar o processo de aquisição de dados e controle do hardware é necessário definir a porta serial a ser utilizada e posteriormente estabelecer a conexão. Ao estabelecer a conexão o sistema envia um comando ao hardware para que o envio de dados para a serial seja interrompido. Isso é necessário pois antes de estabelecer o controle sobre o hardware é necessário verificar se a versão do firmware é igual a versão do projeto. Isso impede de abrir uma conexão serial com outro equipamento qualquer que esteja na lista de portas disponíveis do sistema. Isso também previne erros de funcionamento, caso versões diferentes de hardware e software sejam utilizadas. Se as versões forem iguais a inicialização continua, do contrário, uma mensagem

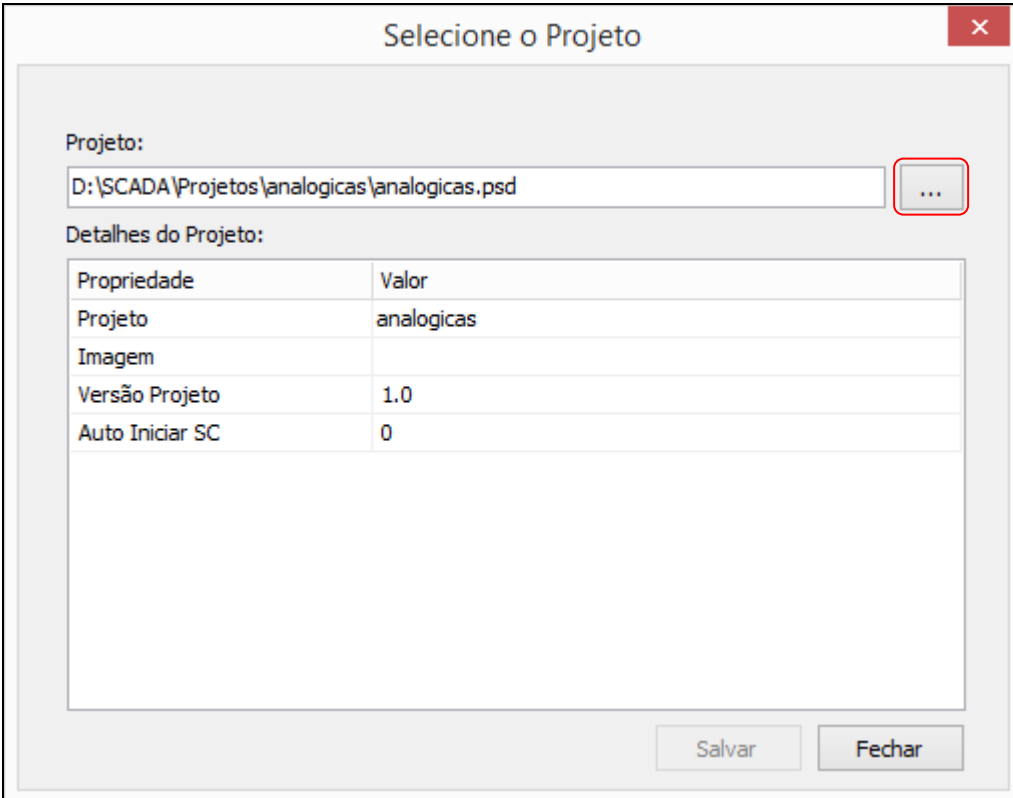
informa ao usuário que as versões de firmware de software e hardware são diferentes, bloqueando a conexão serial.

Após o teste de versão o sistema envia um comando ao hardware para retornar todos os valores das variáveis do tipo escrita. Desta forma os campos são preenchidos com os valores atuais do hardware. Depois de atualizado todos os valores, o sistema envia o comando ao hardware para inicializar a comunicação serial. A partir deste ponto o sistema está conectado ao hardware, toda alteração do hardware é refletida no sistema e as alterações no sistema são aplicadas ao hardware.

3.3 Sistema MANAGER

Caso o usuário tenha diferentes projetos, a seleção do projeto deve ser feita utilizando a ferramenta Manager. Esta aplicação é responsável por definir o arquivo de inicialização do projeto. O arquivo de inicialização do projeto é o SCConfig.ini. A figura 15 exibe a tela da aplicação Manager.

Figura 15 – Tela da aplicação Manager

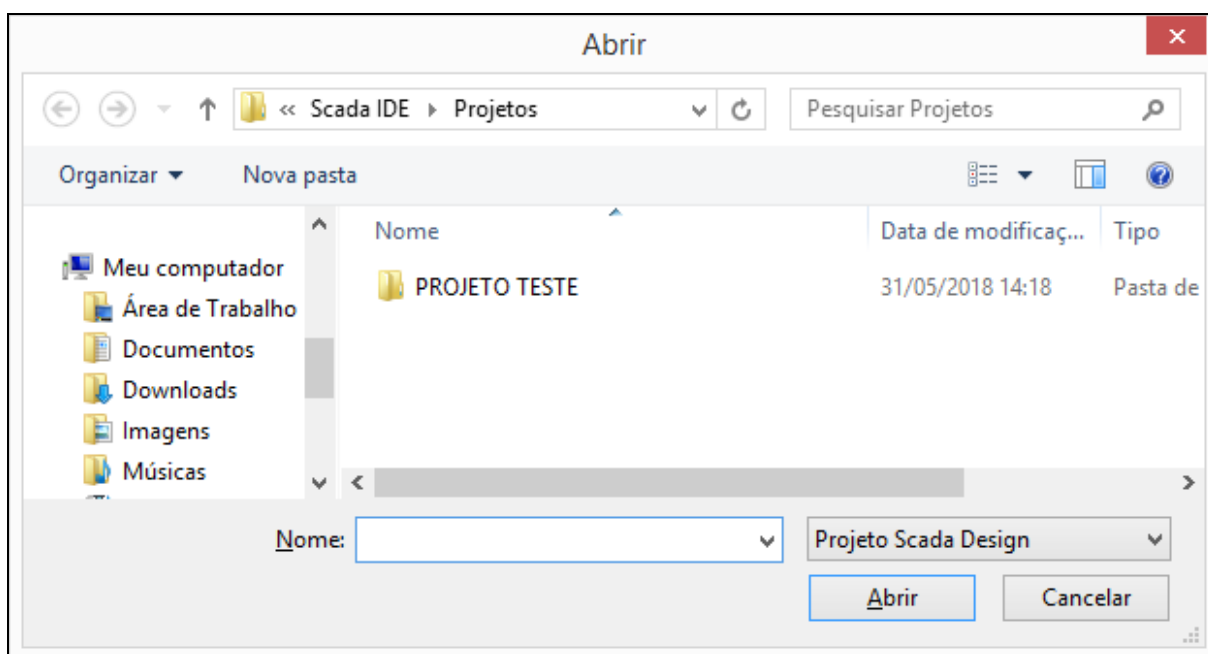


Propriedade	Valor
Projeto	analogicas
Imagem	
Versão Projeto	1.0
Auto Iniciar SC	0

Fonte: Elaborado pelo autor.

Ao selecionar o botão destacado acima, uma caixa de seleção de projetos é aberta, conforme exibe a figura 16.

Figura 16 – Janela de seleção de projeto



Fonte: Elaborado pelo autor.

Ao selecionar um projeto (arquivo *.psd) os detalhes do projeto são carregados na tela. Clicando no botão Salvar, o arquivo SCConfig.ini é atualizado.

3.4 Sistema RegisterExt

O sistema RegisterExt é responsável por criar uma chave de registro dentro do registro do Windows, criando uma associação para os arquivos *.psd, e desta forma o Windows passa a reconhecer essa extensão como "Projeto Scada Design". Assim ao abrir um arquivo com a extensão *.psd o programa Scada Design abrirá automaticamente. Para que as alterações tenham efeito é necessário reiniciar o computador ou o processo "explorer".

4 CARACTERÍSTICAS DO FIRMWARE

Para facilitar o processo de integração entre o programa do micro controlador (*.ino) e as aplicações de Design e Controle foi criado o arquivo “exemplo.ino” que se encontra dentro do diretório “C:\Program Files (x86)\Scada IDE\Arduino\exemplo”. Ele será utilizado como base para as explicações dos parágrafos seguintes.

4.1 Programação do Firmware

O firmware para ser interpretado de maneira correta pelo sistema supervisório (*Scada Control*) dever seguir algumas regras de escrita. Desta forma quando o projeto for criado, e o recurso de “Importar Configurações do Arquivo” for utilizado, os nomes das variáveis e pinos de entrada e saída serão importados corretamente.

4.2 Pinos de entrada ou saída

Devem ser definidos dentro da sessão de definição de variáveis. Por segurança, estes pinos são definidos com o modificador const (de constante). Desta forma, o compilador não permite alterar o valor destas variáveis.

Sintaxe de definição do nome do pino de entrada ou saída:

```
const TIPO NOME_DO_PINO = NUMERO DO PINO;
```

Exemplo:

```
const int pinoEntrada = 5;
```

```
const int pinoSaida = 6;
```

Neste exemplo, definimos duas variáveis do tipo inteiro. Ainda não foi definido o modo de operação, ou seja, se ele será utilizado como entrada ou saída digital.

Após a declaração da variável, vamos definir o modo do pino dentro da sessão setup.

A sintaxe de definição dos pinos dentro da sessão setup é:

```
pinMode(NOME_DO_PINO, MODO);
```

Sendo o MODO:

- OUTPUT: determina que o pino vai ser utilizado com saída digital;

- INPUT_PULLUP: determina que o pino vai ser utilizado como entrada digital;

As definições para os pinos do nosso exemplo acima resulta em:

```
pinMode(pinoEntrada, OUTPUT);
```

```
pinMode(pinoSaida, OUTPUT);
```

A extração do nome do pino ocorre de forma sequencial, varrendo o conteúdo do arquivo. Na ocorrência da palavra pinMode, o algoritmo separa a string contida dentro dos parênteses em duas utilizando o delimitador vírgula. O primeiro segmento diz respeito ao nome da variável e o segundo ao modo (saída ou entrada).

4.3 Entradas Analógicas

Assim como na definição dos pinos de entrada e saída, a definição do nome da variável que vai receber os dados da leitura analógica é definida na sessão definição de variáveis.

Sintaxe de definição do nome da variável de leitura analógica:

TIPO NOME_VARIAVEL = VALOR INICIALIZADO;
--

Exemplo:

```
int valorA0 = 0;
```

Neste exemplo declaramos uma variável do tipo inteiro com o nome valorA0 e inicializamos com valor zero.

Dentro da sessão loop, é definido a leitura analógica. A sintaxe da leitura analógica é:

NOME_VARIAVEL = analogRead(PINO_ENTRADA);

Onde:

- NOME_VARIAVEL: nome da variável definida na sessão definição de variáveis;
- PINO_ENTRADA: pino de entrada analógica. Verifique o modelo de placa utilizada. Para este projeto exemplo foi utilizado uma placa modelo UNO que conta com 6 entradas analógicas, enumeradas de A0 (analógica 0) até A5 (analógica 5);

O micro controlador trabalha com um conversor analógico/digital com resolução de 10 bits. Desta forma o valor retornado para a função estará entre 0 e 1023. Para tratar este sinal, colocando-o dentro de uma faixa desejável, utiliza-se a função map. A função recebe como parâmetros a entrada analógica, o valor inferior da entrada analógica, o valor superior da entrada analógica, o valor inferior desejado, e o valor superior desejado.

Exemplo: `valorA0 = map(analogRead(A0), 0 , 1023, 0, 100);`

Neste exemplo, as leituras da função vão retornar valores entre 0 e 1023, mas a função map vai atribuir uma nova escala entre 0 e 100. Assim quando a entrada analógica tiver o seu valor mínimo a leitura será 0 e quando o valor da entrada analógica for o máximo, o seu valor será 100.

Para o processamento do nome das variáveis analógicas, o algoritmo procura pela ocorrência da palavra `analogRead`. Ao encontrar, o conteúdo antes do sinal de igual indica o nome da variável, e o seu tipo é identificado pelo conteúdo entre sinal de parênteses após a palavra `analogRead`.

4.4 Interpretador de comandos do micro controlador

O micro controlador opera com quatro comandos definidos em seu algoritmo. São eles o `cmd`, `get`, `set` e `out`. Ao receber uma mensagem pela serial, ele separa a string utilizando o delimitador de dois pontos. O primeiro campo corresponde ao prefixo do comando, o segundo campo ao nome da variável e o terceiro campo, quando aplicável, ao valor da variável.

O comando `cmd` é responsável por enviar um comando para o firmware. A sintaxe do comando `cmd` é:

`cmd:NOME_VARIAVEL:VALOR`

O comando `get` é responsável por retornar um valor de uma variável do firmware. A sintaxe do comando `get` é:

`get:NOME_VARIAVEL`

O comando `set` é responsável por atribuir um novo valor a uma variável dentro do firmware. A sintaxe do comando `set` é:

`set:NOME_VARIAVEL:NOVO_VALOR`

O comando out é responsável por atuar sobre uma saída digital. A sintaxe do comando out é:

```
out:NOME_VARIAVEL:VALOR
```

Onde o valor é um valor binário (0 ou 1).

Para demonstrar o uso dos comandos, vamos utilizar analisar o código projeto exemplo.

Como condição inicial de operação do micro controlador, o fluxo de dados para a porta serial é desabilitado. Uma variável booleana (startPrint) sempre inicializa na condição “false”. Este comando é habilitado apenas após o sistema supervisorio estabelecer a conexão e verificar se as versões de firmware são compatíveis, e todos os campos do tipo escrita tenham seus valores atualizado com os do micro controlador. Assim, quando o supervisorio estabelece um pedido de conexão com o hardware, o sistema envia o comando:

```
cmd:startPrint:0
```

Desta forma, o sistema força a interrupção do envio de dados. Após ocorre o envio do comando de solicitação da versão do firmware do hardware.

```
get:firmware
```

Ao executar o comando get, a serial envia a mensagem de retorno com o formato e conteúdo abaixo:

```
get:firmware:1.4
```

Deste modo o interpretador de comandos do sistema supervisorio processa a mensagem separando a string pelo delimitador dois pontos (“:”). Assim, ele verifica a condição da versão do firmware do hardware ser igual a versão do software. Após este teste, se o resultado for verdadeiro, a próxima função a ser executada é a solicitação dos valores de todos as variáveis correspondentes aos componentes do tipo escrita do sistema supervisorio. Isso é realizado para atualizar o valor dos campos para os valores correntes do hardware. Sendo assim é extremamente importante que o usuário controle as versões de firmware do programa do micro controlador e do projeto Scada Design.

O controle dentro do firmware é feito na linha de código abaixo:

```
const String firmware = "1.0";
```

A figura 17 exibe o segmento de código que contém o campo responsável por controlar a versão do firmware.

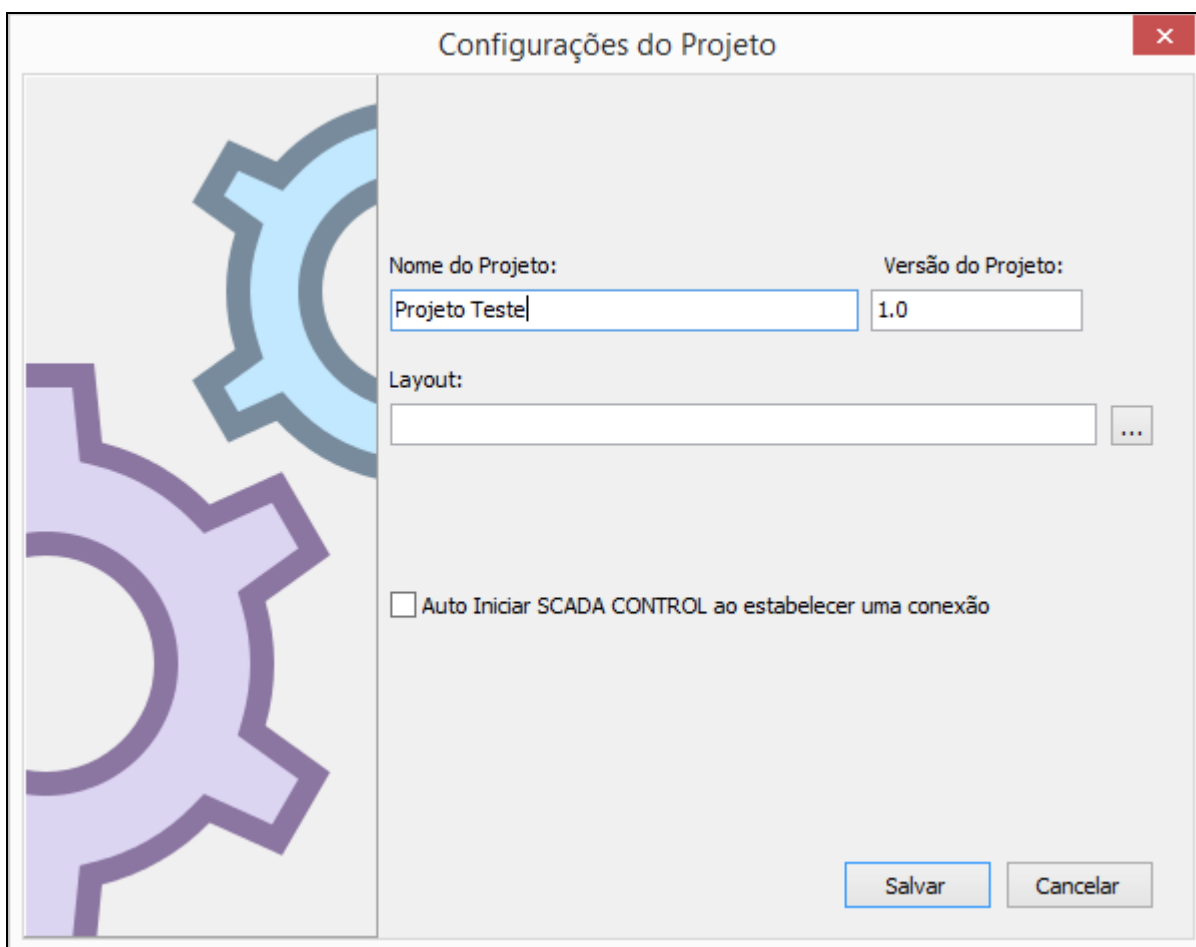
Figura 17 – Campo de definição da versão do firmware

```
template
//=====
//      INFORMAÇÕES DO PROJETO
//=====
const String autor = "Giliardi Oliveira";
const String firmware = "1.0";
const String data = "05/05/18";
```

Fonte: Elaborado pelo autor.

E o controle dentro do projeto Scada Design é feito acessando o menu de configurações do projeto, conforme exibe a figura 18.

Figura 18 – Janela de Configurações do Projeto



Configurações do Projeto

Nome do Projeto: Projeto Teste

Versão do Projeto: 1.0

Layout:

☐ Auto Iniciar SCADA CONTROL ao estabelecer uma conexão

Salvar Cancelar

Fonte: Elaborado pelo autor.

Quando ocorre a inicialização do sistema supervisorio, uma lista dos componentes do tipo escrita é instanciada em memória. Com base nesta lista o supervisorio executa um laço retornando a propriedade link do componente (corresponde ao nome da variável) para o comando get. Assim, o número de comandos get enviados será igual ao número de componentes do tipo escrita existentes na lista. A sintaxe do comando será:

```
get:NOME_VARIAVEL
```

Após retornar todos os componentes do tipo escrita, um novo comando é enviado ao hardware, alterando a flag startPrint para true.

```
set:startPrint:1
```

Logo após o envio deste comando, o hardware inicia a transmissão de dados para a serial ininterruptamente.

O comando out está associado somente ao componente botão, e é executado sempre que ocorrer um clique sobre o botão. Ao clicar sobre o botão o comando executado será:

```
out:NOME_VARIAVEL:VALOR
```

Exemplo: out:pinoSaida:0

Este comando desliga a saída correspondente ao pinoSaida.

4.5 Envio de dados da serial para o Scada Control

O envio de dados pela serial ocorre com o uso do comando sprintf. Para que os dados sejam interpretados pelo sistema supervisorio, devem obedecer a sintaxe:

```
sprintf (buffer, "#2NOME_VARIAVEL:%d#3", NOME_VARIAVEL);
```

Sendo:

buffer = array do tipo char;

#2 = marcador de início da mensagem;

#3 = marcador de fim da mensagem;

NOME_VARIAVEL = nome da variável do tipo inteiro. Também podem ser enviados valores do tipo boolean, pois vão resultar em valores 0 ou 1. Isso é útil para a leitura de entradas digitais.

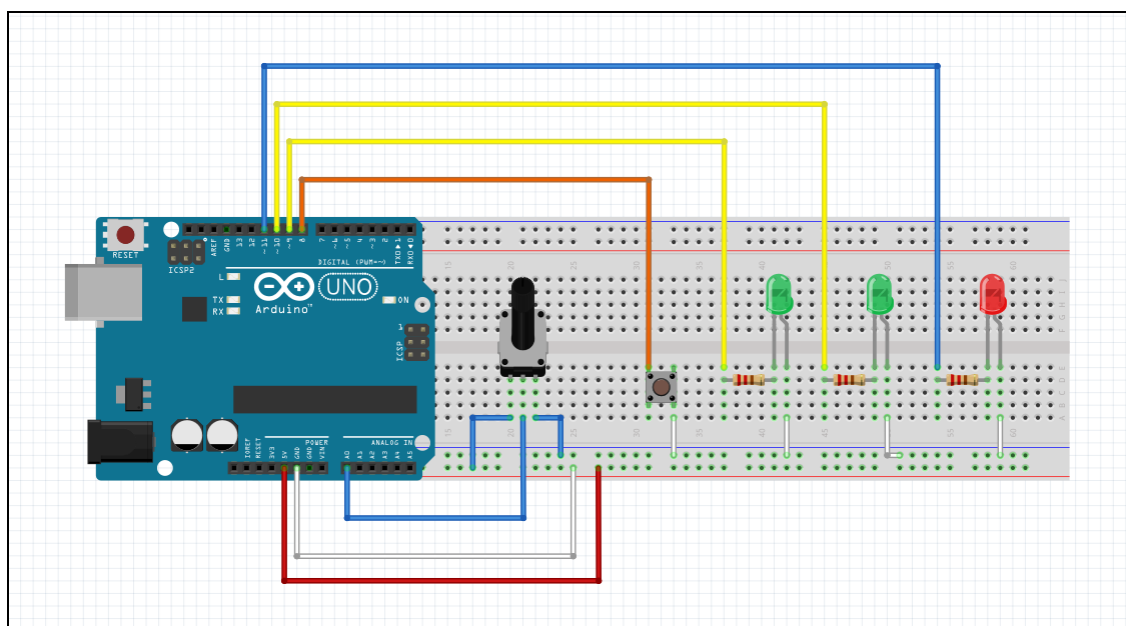
Após ocorre o envio dos dados para a serial, seguido de um comando de delay.

```
Serial.println(buffer);  
delay(10);
```

5 PROJETO EXEMPLO

Caso o usuário tenha interesse em montar o hardware conforme o projeto exemplo que acompanha a instalação, faça a montagem de acordo com o diagrama da figura 19.

Figura 19 – diagrama do projeto exemplo



Fonte: Elaborado pelo autor.