# Resumo do Projeto - Sistema de Controle de Bobinadeira

### Controle CFW500 via Modbus RTU



## Sumário Executivo

Este projeto implementa um sistema completo de controle e monitoramento para uma máquina bobinadeira com três rolos independentes, cada um controlado por um inversor de frequência WEG CFW500. O sistema utiliza comunicação Modbus RTU entre um CLP Allen Bradley Micro850 e os inversores, com interface de operação através de um IHM PanelView 800.

#### Dados Rápidos do Projeto

Item	Descrição
Aplicação	Sistema de Controle de Bobinadeira Industrial
Inversores	3× WEG CFW500 (Nós Modbus 1, 2, 3)
CLP	Allen Bradley Micro850
IHM	Allen Bradley PanelView 800
Protocolo	Modbus RTU
Linguagem	IEC 61131-3 Structured Text (ST)
Versão Atual	2.2
Data	Outubro 2025



## **©** Objetivos do Sistema

### Objetivo Principal

Controlar automaticamente a velocidade e torque de três rolos em uma bobinadeira, garantindo:

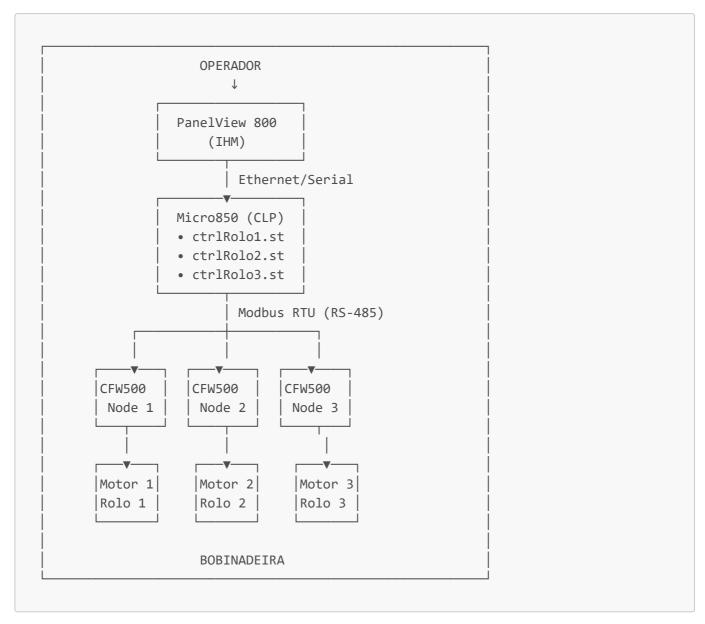
- Tensão constante no material
- Sincronização entre rolos
- Proteção contra sobrecarga
- Operação segura e confiável

#### **Objetivos Específicos**

- 1. **Controle de Velocidade**: Ajuste preciso de velocidade mínima e máxima por rolo (0-100 Hz)
- 2. **Controle de Torque**: Limitação de torque máximo por rolo (0-200%)
- 3. Monitoramento em Tempo Real: Leitura contínua de velocidade, torque e corrente
- 4. Detecção de Falhas: Identificação automática de problemas de comunicação e operação

- 5. **Alarme Preventivo**: Detecção de condições anormais (velocidade vs torque)
- 6. Interface Amigável: Operação simples via IHM touchscreen

# Arquitetura do Sistema



# Estrutura de Arquivos do Projeto

Arquivos de Código (Structured Text)

Arquivo	Linhas	Descrição
ctrlRolo1.st	~450	Controle completo do Rolo 1 (Node 1)
ctrlRolo2.st	~450	Controle completo do Rolo 2 (Node 2)
ctrlRolo3.st	~450	Controle completo do Rolo 3 (Node 3)

Total de Código: ~1.350 linhas de ST

#### Arquivos de Documentação (Markdown)

Arquivo	Páginas	Descrição
README.md	8	Visão geral e guia de início rápido
variaveisGlobais.md	12	Todas as variáveis do sistema
CFW500_Modbus_Mapping.md	15	Mapeamento Modbus (inglês)
CFW500_Mapeamento_Modbus_Portugues.md	15	Mapeamento Modbus (português)
IHM_Variaveis_PanelView800.md	18	Especificação do IHM
RESUMO_PROJETO.md	12	Este documento

Total de Documentação: ~80 páginas

## Funcionalidades Principais

#### 1. Controle de Parâmetros

#### Setpoints Ajustáveis (por Rolo)

- Velocidade Máxima (0.0 100.0 Hz)
- **Velocidade Mínima** (0.0 100.0 Hz)
- **Torque Máximo** (0.0 200.0%)

#### Validações Automáticas

- Verificação de limites (clamping automático)
- SpeedMin ≤ SpeedMax (garantido)
- ✓ Conversão de unidades (Hz → 0.01Hz, % → 0.1%)

#### 2. Monitoramento em Tempo Real

#### Variáveis Monitoradas (por Rolo)

- Torque Atual (P0009) Atualização: 10s
- S Velocidade Atual (P0002) Atualização: 10s
- Corrente de Saída (P0003) Atualização: 10s
- Parâmetro P0409 Atualização: 10s
- 🖸 Parâmetro P0410 Atualização: 10s

#### **Escalamento Automático**

- Raw UINT → REAL escalado
- Exemplo: 1234 (UINT) → 123.4 (REAL) para torque

#### 3. Verificação Automática de Parâmetros

#### Ciclo de Verificação (a cada 10 segundos)

- 1. ✓ **P0100** (Tipo de Controle) → Força para 2 (Vetorial Sensorless)
- 2. ✓ P0134/P0133 (Velocidades) → Força valores configurados
- 3. ✓ **P0169/P0170** (Torques) → Força valores configurados

Propósito: Garante que parâmetros críticos não sejam alterados externamente

4. Sistema de Alarmes

#### Alarme de Velocidade vs Torque

- Condição: Torque > 20% AND Velocidade < (Torque × 0.5)
- **Delay**: 5 segundos antes de ativar
- Indicação: SpeedTorqueAlarm\_roloX = TRUE
- Causa Provável: Travamento do motor ou potência insuficiente

#### Alarmes de Comunicação

- **WriteError**: Falha ao escrever parâmetros (3 tentativas)
- ReadError: Falha ao ler parâmetros (3 tentativas)
- CommunicationOK: Status geral da comunicação Modbus

#### 5. Sistema de Retry e Recuperação

#### Mecanismo de Retry

- **Tentativas**: Até 3 por operação
- Contadores Independentes:
  - Write operations
  - Read operations (P0009, P0002, P0003)
  - Read operations (P0409, P0410)
  - Verification operations

#### Recuperação Automática

- **Tempo**: 30 segundos de comunicação bem-sucedida
- Ação: Limpeza automática de flags de erro
- Objetivo: Evitar alarmes persistentes após falhas temporárias

#### 6. Arbitragem de Barramento Modbus

#### Flag Global: ModbusBusy

- Compartilhado entre os 3 rolos
- **Previne**: Operações Modbus simultâneas
- Garante: Acesso sequencial ao barramento

#### Sequenciamento

```
Rolo1: Verifica ModbusBusy

Se FALSE → Set TRUE → Executa operação → Set FALSE

Rolo2: Aguarda ModbusBusy = FALSE

Rolo3: Aguarda ModbusBusy = FALSE
```

#### 7. Detecção Inteligente de Mudanças

#### **Escrita Sob Demanda**

- Detecta mudanças nos setpoints do IHM
- Compara com valores atuais do inversor
- Só escreve se houver diferença real
- Vantagem: Reduz tráfego Modbus e desgaste de memória EEPROM

## Ciclos de Operação

#### Ciclo de Escrita (Sob Demanda)

```
TRIGGER: Mudança no IHM ou botão "Aplicar"

↓
STEP 0: Escreve P0134 (Velocidade Máxima)

↓
STEP 1: Escreve P0133 (Velocidade Mínima)

↓
STEP 2: Escreve P0169 (Torque Máximo)

↓
STEP 3: Escreve P0170 (Torque Mínimo)

↓
STEP 4: Escreve P0498 (Salvar)

↓
FIM: Reset triggers, volta ao STEP 0
```

#### Ciclo de Leitura (A cada 10s)

```
TIMER: 10 segundos

↓

STEP 1: Lê P0009 (Torque)

↓

STEP 2: Lê P0002/P0003 (Velocidade/Corrente)

↓

├─→ Calcula SpeedTorqueAlarm

↓

FIM: Volta ao STEP 0, aguarda próximo timer
```

### Ciclo P0409/P0410 (A cada 10s - Separado)

```
TIMER: 10 segundos (independente)
↓
STEP 1: Lê P0409/P0410 (consecutivos)
↓
FIM: Volta ao STEP 0, aguarda próximo timer
```

### Ciclo de Verificação (A cada 10s)

```
TIMER: 10 segundos

↓

STEP 1: Lê P0100 → Se ≠ 2, escreve 2

↓

STEP 2: Lê P0134/P0133 → Se ≠ setpoints, força valores

↓

STEP 3: Lê P0169/P0170 → Se ≠ setpoints, força valores

↓

FIM: Volta ao STEP 0, aguarda próximo timer
```

## 

#### Resumo Quantitativo

Categoria	Quantidade por Rolo	Total (3 Rolos)
Entrada (REAL)	3	9
Monitoramento (REAL)	5	15
Status/Alarme (BOOL)	5	15
Controle (BOOL)	2	6
Internas (UINT)	~20	~60
Timers (TON)	4	12
MSG_MODBUS	6	18
TOTAL	~45	~135

Variáveis Críticas

Por Rolo (Exemplo: Rolo 1)

#### **Entrada (Operador** → **CLP)**:

SpeedMax\_rolo1 (REAL)

- SpeedMin\_rolo1 (REAL)
- TorqueMax\_rolo1 (REAL)

#### **Monitoramento (CLP → Operador)**:

- TorquePercentScaled\_rolo1 (REAL)
- OutputFreqScaled\_rolo1 (REAL)
- OutputCurrentScaled\_rolo1 (REAL)
- P0409Scaled rolo1 (REAL)
- P0410Scaled\_rolo1 (REAL)

#### Status:

- CommunicationOK\_rolo1 (BOOL)
- WriteError\_rolo1 (BOOL)
- ReadError\_rolo1 (BOOL)
- SpeedTorqueAlarm\_rolo1 (BOOL)
- EnableDrive\_rolo1 (BOOL)



# Parâmetros Modbus CFW500

### Parâmetros Principais

Parâmetro	Registro	Função	Acesso	Escala
P0002	2	Frequência de Saída	R	×0.1 Hz
P0003	3	Corrente de Saída	R	×0.1 A
P0009	9	Torque do Motor	R	×0.1 %
P0100	100	Tipo de Controle	R/W	1:1
P0133	133	Velocidade Mínima	R/W	×0.01 Hz
P0134	134	Velocidade Máxima	R/W	×0.01 Hz
P0169	169	Torque Máximo	R/W	×0.1 %
P0170	170	Torque Mínimo	R/W	×0.1 %
P0409	409	Adicional 1	R	Variável
P0410	410	Adicional 2	R	Variável
P0498	498	Salvar Parâmetros	R/W	0/1

### Códigos de Função Modbus

- FC 3: Read Holding Registers (leitura)
- FC 6: Write Single Register (escrita individual)
- FC 16: Write Multiple Registers (escrita múltipla)

# Interface IHM

### Telas Planejadas

#### 1. Tela Principal (Dashboard)

- Visão geral dos 3 rolos
- o Valores em tempo real
- o Indicadores de status
- Navegação para telas detalhadas

#### 2. **Tela de Configuração** (3× - uma por rolo)

- o Campos numéricos editáveis
- o Botões +/- para ajuste fino
- o Displays de monitoramento
- Controles de habilitação
- o Botão "Aplicar" (aciona WriteTrigger)

#### 3. Tela de Alarmes

- Status detalhado de cada rolo
- Descrição de alarmes ativos
- Histórico de eventos
- Botões de reconhecimento

#### 4. Tela de Configurações Globais

- o Informações do sistema
- o Limites de segurança
- o Parâmetros de comunicação

#### Elementos de Interface

- 45 variáveis vinculadas
- Cores padronizadas: Verde (OK), Amarelo (Aviso), Vermelho (Erro)
- Atualização: 1-2 segundos no IHM (dados do CLP atualizados a cada 10s)

# Segurança e Confiabilidade

### Mecanismos de Segurança

#### 1. Validação de Entrada

- o Limites de velocidade: 0-100 Hz
- Limites de torque: 0-200%
- SpeedMin ≤ SpeedMax garantido

### 2. **Proteção de Comunicação**

Retry automático (3 tentativas)

- Timeout configurável
- o Arbitragem de barramento

### 3. Verificação Periódica

- Parâmetros críticos verificados a cada 10s
- o Correção automática se alterados externamente

#### 4. Alarmes Preventivos

- Detecção de travamento (velocidade vs torque)
- o Delay de 5s para evitar falsos alarmes

### 5. **Recuperação Automática**

- o Limpeza de erros após 30s de operação normal
- o Evita alarmes persistentes

#### Redundâncias

- Validação no CLP e no IHM
- Múltiplos contadores de retry
- Flags de erro separadas por tipo de operação

## Performance e Otimização

### Temporização

Operação	Frequência	Justificativa
Leitura P0009/P0002/P0003	10s	Processo lento, não requer atualização rápida
Leitura P0409/P0410	10s	Parâmetros auxiliares
Verificação	10s	Garantia de integridade de parâmetros
Escrita	Sob demanda	Apenas quando operador altera
Retry	Imediato	Maximiza chance de sucesso

#### Otimizações Implementadas

- 1. Leitura Consecutiva: P0002/P0003 e P0409/P0410 lidos juntos (1 transação)
- 2. Escrita Sob Demanda: Só escreve quando valores mudam
- 3. Arbitragem Global: Evita colisões no barramento Modbus
- 4. Ciclos Independentes: 4 ciclos separados não interferem entre si

# Q Diagnóstico e Manutenção

### Pontos de Verificação

#### **Operação Normal**

- CommunicationOK\_roloX = TRUE
- WriteError\_roloX = FALSE
- ReadError\_roloX = FALSE
- SpeedTorqueAlarm\_roloX = FALSE
- Valores de monitoramento atualizando

#### **Problemas Comuns**

Sintoma	Causa Provável	Solução
CommunicationOK = FALSE	Cabo RS-485 desconectado	Verificar fiação
WriteError = TRUE	Inversor em modo local	Mudar para modo remoto
ReadError = TRUE	Endereço Modbus incorreto	Verificar Node ID
SpeedTorqueAlarm = TRUE	Motor travado	Verificar carga mecânica
Valores não atualizam	Timer não funcionando	Verificar scan do CLP

### Logs e Rastreabilidade

- Todos os erros têm flags dedicadas
- Contadores de retry indicam qualidade de comunicação
- Histórico de alarmes no IHM (se implementado)

## **邑** Documentação Disponível

#### **Documentos Técnicos**

- 1. **README.md** Guia de início rápido
- 2. variaveisGlobais.md Lista completa de variáveis
- 3. CFW500\_Modbus\_Mapping.md Mapeamento Modbus (EN)
- 4. CFW500\_Mapeamento\_Modbus\_Portugues.md Mapeamento Modbus (PT)
- 5. IHM\_Variaveis\_PanelView800.md Especificação IHM completa
- 6. RESUMO\_PROJETO.md Este documento

#### Referências Externas

- Manual WEG CFW500 Parâmetros
- Manual WEG CFW500 Comunicação Modbus
- Allen Bradley Micro850 User Manual
- Allen Bradley PanelView 800 Configuration Guide
- IEC 61131-3 Structured Text Programming



## Próximos Passos / Expansões Futuras

### Funcionalidades Sugeridas

#### 1. Controle de Receitas

- o Salvar/carregar conjuntos de parâmetros
- Receitas por tipo de material

#### 2. Datalog

- Registro histórico de velocidade, torque, corrente
- o Exportação para análise

#### 3. Parâmetros Adicionais

- o P0001 (Velocidade do Motor em RPM)
- P0007 (Temperatura do Inversor)
- P0220/P0221 (Tempos de aceleração/desaceleração)

#### 4. Sincronização Automática

- o Ajuste automático de velocidades entre rolos
- o Controle de tensão do material

#### 5. Manutenção Preditiva

- Contadores de horas de operação
- o Alertas de manutenção preventiva

#### 6. Comunicação Ethernet

- o Migrar de Modbus RTU para Modbus TCP
- Acesso remoto via web

## Pontos Fortes do Projeto

#### **Técnicos**

Código Modular: Cada rolo com programa independente ✓ Documentação Completa: >80 páginas de documentação ✓ Tratamento de Erros: Retry + recuperação automática ✓ Validação: Múltiplas camadas de verificação ✓ Escalabilidade: Fácil adicionar novos parâmetros ✓ Manutenibilidade: Código comentado e bem estruturado

#### Operacionais

✓ Interface Amigável: IHM touchscreen intuitivo ✓ Segurança: Limites e validações automáticas ✓ Confiabilidade: Sistema de retry robusto ✓ Diagnóstico: Alarmes claros e informativos ✓ Flexibilidade: Parâmetros ajustáveis em tempo real



### Estrutura do Código

• Linguagem: IEC 61131-3 Structured Text

• Ambiente: Connected Components Workbench (CCW)

• Padrão: Código segue boas práticas IEC

#### Contatos Técnicos

• **Documentação**: Consultar arquivos .md neste diretório

• Mapeamento Modbus: CFW500\_Modbus\_Mapping.md

• Variáveis: variaveisGlobais.md

• IHM: IHM\_Variaveis\_PanelView800.md



## Histórico de Versões

Versão	Data	Mudanças Principais
1.0	Out/2025	Implementação inicial
2.0	Out/2025	+ Verificação automática, retry logic, alarmes
2.1	Out/2025	Correção de periodicidade dos timers (10s)
2.2	Out/2025	+ P0409/P0410, ciclo independente, documentação completa



## Status do Projeto

### Completude

Item	Status	Percentual
Código PLC	Completo	100%
Documentação	Completa	100%
Mapeamento Modbus	Completo	100%
Especificação IHM	Completa	100%
Telas IHM	<b>Pendente</b>	0%
Testes em Bancada	<b>Pendente</b>	0%
Comissionamento	▼ Pendente	0%

### Próximas Ações

- 1. Z Desenvolver telas no FactoryTalk View Studio
- 2. Testar comunicação Modbus em bancada
- 3. X Validar lógica de alarmes
- 4. Realizar testes integrados
- 5. T Commissioning no equipamento final

### **⊘** Conclusão

Este projeto representa uma solução completa e profissional para controle industrial de bobinadeiras. Com código robusto, documentação extensiva e interface planejada, o sistema está pronto para implementação e oferece base sólida para expansões futuras.

#### **Destaques Finais**

- ~1.350 linhas de código ST de alta qualidade
- ~80 páginas de documentação técnica
- 45 variáveis por rolo para controle total
- 4 ciclos independentes para máxima eficiência
- Retry automático para confiabilidade
- Alarmes inteligentes para operação segura

Documento gerado em: Outubro 2025 Versão do Sistema: 2.2 Status: Pronto para Implementação de IHM