

Disciplina: Microcontroladores - Engenharia da Computação - UEMA

Entregas: N2 (Simulação) + N3 (Implementação Prática)

Projeto Único - Trabalhado em Equipes de Até 5 Alunos

VISÃO GERAL DO PROJETO

Desenvolver um sistema de simulação de medição e calibração didática de máquinas de solda SMAW (eletrodo) e MIG/MAG, baseado em Arduino, capaz de:

- ✓ medir grandezas elétricas e mecânicas
- ✓ apresentar leituras em displays locais
- ✓ comunicar-se com um supervisor em Python
- ✓ registrar dados, gerar gráficos, alarmes e relatório final em PDF

Esse sistema simula um sistema de medição e calibração usado em máquinas de solda SMAW (eletrodo) e MIG/MAG, em ambiente seguro, com medições equivalentes.

✓ FASE 1 (N2) — SIMULAÇÃO COMPLETA NO PROTEUS (04/12)

Entrega:

- ✓ Arquivo Proteus (.pdsprj)
- ✓ Código Arduino (.ino)
- ✓ Código/Projeto do Supervisor
- ✓ Supervisor em Python (Tkinter ou outra GUI)
- ✓ CSV + PDF gerado
- ✓ Guia do usuário

1) SIMULAÇÃO DAS GRANDEZAS (OBRIGATÓRIO NO PROTEUS)

- A) Tensão de saída (V_{avg} , V_{rms} , V_{pk}) – 0 a 100 V equivalentes
- B) Corrente de saída (I_{avg} , I_{rms} , I_{pk}) – 0 a 500 A equivalentes
- C) RPM da alimentação do arame
- D) Vazão de gás (SIMULADA – L/min)
- E) Temperatura (°C)
- F) Estados dos Relés (5 relés)

2) IHM LOCAL OBRIGATÓRIA (NO PROTEUS)

- Displays obrigatórios: Dois módulos 7 segmentos × 4 dígitos

Um exibe a **Tensão (V)**

Outro exibe a **Corrente (A)**

- LCD 16×2

Deve mostrar: RPM , Vazão, Temperatura, Modo atual (SMAW / MIG) , Erros e alarmes

Botões físicos na IHM: MENU, UP, DOWN, ENTER

3) Funções mínimas da IHM:

- Seleção de modo (SMAW/MIG)
- Acesso ao menu de calibração
- Leitura local (online)
- Indicação de alarmes com ACK

A) Conexão

- Botão **Iniciar/Comunicar**
- Exibir status **ONLINE/OFFLINE**



B) Controles

- 5 botões digitais → acionam 5 relés no Proteus
- 2 sliders analógicos → enviam valores ao Arduino (para validações)

✓ C) Visualizações

- **3 gráficos** em tempo real:
 - Tensão × tempo
 - Corrente × tempo
 - RPM/Vazão × tempo
- **5 gauges**:
 - Tensão
 - Corrente
 - RPM
 - Vazão
 - Temperatura

✓ D) Gravação e Relatório

- Botão “**Gravar CSV**”
- Botão “**Gerar PDF**” contendo:
 - Gráficos
 - Tabelas (médias, RMS, picos)
 - Estados dos relés
 - Alarmes ocorridos
 - Resultado final (APROVADO/REPROVADO)

✓ E) Alarmes

- Sobretenção
- Sobrecorrente
- Baixa vazão
- Temperatura alta
- Falha de comunicação
- Botão **ACK** deve limpar o alarme

4) SUPERVISÓRIO EM PYTHON (OBRIGATÓRIO)

- Tkinter , Tkinter + Matplotlib , PyQt5, CustomTkinter , NiceGUI , Node-Red

FUNCIONALIDADES OBRIGATÓRIAS DO SUPERVISÓRIO

A) Conexão

- Botão Iniciar/Comunicar
- Exibir status ONLINE/OFFLINE

B) Controles

- 5 botões digitais → acionam 5 relés no Proteus
- 5 sliders analógicos → enviam valores ao Arduino (para validações)

C) Visualizações

- 3 gráficos em tempo real:

Tensão × tempo

Corrente × tempo

RPM/Vazão × tempo

- 5 gauges: Tensão , Corrente , RPM , Vazão, Temperatura

D) Gravação e Relatório

- Botão “Gravar CSV”
- Botão “Gerar PDF” contendo: Gráficos, Tabelas (médias, RMS, picos), Estados dos relés, Alarmes ocorridos, Resultado final (APROVADO/REPROVADO).

E) Alarmes: Sobretensão, Sobrecorrente, Baixa vazão, Temperatura alta, Falha de comunicação, Botão ACK deve limpar o alarme.

5) CALIBRAÇÃO (OBRIGATÓRIO)

A calibração deve ser simulada no Proteus e realizada no supervisorio.

A) IZERO — Zero da corrente

- POT em 0 V (simulando 0 A)
- Supervisorio coleta 2 s → calcula **offset**

B) ISPAN — Ganho da corrente

- Ajustar POT para valor conhecido
- Inserir valor no Supervisorio ("I_ref")
- Calcular **K_I**

C) VSPAN — Ganho da tensão

- POT em 0 V → medir
- POT em valor alto (80–100 V equivalente) → medir
- Supervisorio calcula **K_V**

D) Armazenamento em EEPROM

O Arduino deve salvar:

- Vzero_I
- K_I
- K_V

5) PERFIS DE OPERAÇÃO (OBRIGATÓRIO)

Perfil SMAW (Eletrodo)

Fase 1: Ignição (pico de corrente, queda de tensão)

Fase 2: 3 degraus de carga

Fase 3: Resfriamento (tudo OFF, ventilação simulada)

O supervisor deve registrar:

- Vavg, Vrms, Vpk
 - Iavg, Irms, Ipk
 - ΔV entre degraus
 - desvio padrão (estabilidade)
-

Perfil MIG/MAG

Fase 1: Pré-fluxo (vazão > 0, rpm = 0)

Fase 2: Partida (rampa de rpm)

Fase 3: Regime MIG (ripple, curtos)

Fase 4: Pós-fluxo

Medições obrigatórias:

- Vavg, Vrms, Vpk
- Iavg, Irms, Ipk
- número de picos (cursos simulados)
- rpm estabilizado
- vazão estabilizada

6) CRITÉRIOS DE AVALIAÇÃO (CALIBRAÇÃO)

Aprovado se:

- Erro $\leq \pm 2\%$
- Estabilidade (desvio padrão) $\leq 3\%$
- Alarmes funcionam e registram
- Perfis executam todas as fases

FASE 2 (N3) — IMPLEMENTAÇÃO PRÁTICA (11/12)

Ambiente físico:

- Fonte regulável de bancada (Tensão e Corrente)
- Divisores para medir tensões da fonte
- Resistores de potência (até 50 W)
- Ventilador para medir RPM
- Potenciômetro simulando vazão
- Sensor LM35 real

O que implementar:

- Parte física equivalente ao Proteus
- Arduino lendo tudo real
- Supervisório funcionando igual
- Geração de CSV e PDF funcionando
- Execução dos perfis SMAW e MIG com valores adaptados

Objetivo:

Reproduzir fisicamente o comportamento da simulação, em baixa potência.

COMPONENTES PARA A PARTE FÍSICA

Arduino UNO ou MEGA, Resistores, Ventilador 12 V ou 24V (leitura de RPM por pulso), LM35, Relés , Protoboard, Resistores de potência (simular carga).

Critérios de Avaliação

| ETAPA | CRITÉRIO | DESCRIÇÃO | PONTOS |
|----------|---------------------------------|---|------------|
| 1 | Funcionamento e simulação | Circuito com Arduino, displays, botões, sensores tudo funcionando local (no Proteus ou físico). | 3 |
| 2 | Interface Python / Supervisório | Interface funcional com leitura serial e exibição de todos os dados | 3 |
| 3 | Organização e layout | Circuito e telas bem organizados, todas funcionais, usabilidade coerente. | 1,5 |
| 4 | Documentação / Relatório | Geração do CSV e PDF com resultados, tolerâncias e conclusão (Aprovado/Reprovado). | 1,5 |
| 5 | Apresentação final | Clareza e objetividade na explicação de todo o funcionamento do projeto, bem como participação da equipe. | 1 |