

DOCUMENTAÇÃO TÉCNICA

SISTEMA DE MONITORAMENTO PARA

ARMAZENAGEM DE SOJA

Versão: 2.6

Data: 2025

Base Técnica: Embrapa Soja

Autor: Edson Santana Alves

ÍNDICE

1. VISÃO GERAL DO SISTEMA
2. ARQUITETURA TÉCNICA
3. ESPECIFICAÇÕES DE HARDWARE
4. PARÂMETROS TÉCNICOS EMBRAPA
5. ESTRUTURA DO CÓDIGO
6. SISTEMA DE CLASSIFICAÇÃO
7. PROTOCOLOS DE COMUNICAÇÃO
8. GESTÃO DE ERROS
9. CONFIGURAÇÃO THINGSPEAK
10. FLUXO DE OPERAÇÃO
11. PROCEDIMENTOS DE EMERGÊNCIA

1. VISÃO GERAL DO SISTEMA

Objetivo Principal

Sistema IoT para monitoramento contínuo das condições de armazenamento de soja, baseado nas normas técnicas da Embrapa, prevenindo perdas por desenvolvimento fúngico e deterioração.

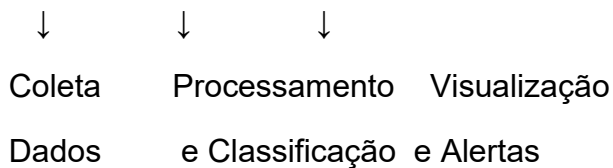
Funcionalidades Principais

- Monitoramento em tempo real de temperatura e umidade
- Classificação automática do risco baseado em thresholds científicos
- Transmissão de dados para plataforma cloud (ThingSpeak)
- Sistema fail-safe e autodiagnóstico
- Alertas visuais para condições críticas

2. ARQUITETURA TÉCNICA

2.1 Diagrama de Blocos

Sensor DHT11 → ESP32-C3 → ThingSpeak Cloud → Dashboard Web



2.2 Componentes Principais

Componente	Especificação	Função
Microcontrolador	ESP32-C3	Processamento central e conectividade
Sensor	DHT11	Medição de temperatura e umidade
Conectividade	Wi-Fi 802.11 b/g/n	Comunicação com cloud
Plataforma Cloud	ThingSpeak	Armazenamento e visualização

Table 1: Componentes principais do sistema

3. ESPECIFICAÇÕES DE HARDWARE

3.1 ESP32-C3

- **CPU:** RISC-V de 32-bit a 160 MHz
- **Memória:** 400 KB SRAM, 4 MB Flash
- **Wi-Fi:** 2.4 GHz 802.11 b/g/n
- **GPIO:** Múltiplos pinos de E/S
- **Alimentação:** 3.3V DC

3.2 Sensor DHT11

- **Temperatura:** -20°C a 60°C (±2°C precisão)
- **Umidade:** 20% a 90% RH (±5% precisão)
- **Taxa de Amostragem:** 1 Hz

- **Interface:** Bus digital de único fio
- **Pino de Dados:** GPIO 4

3.3 Conexões

```
#define DHT_PIN 4 // GPIO4 para comunicação com DHT11
```

4. PARÂMETROS TÉCNICOS EMBRAPA

4.1 Thresholds de Temperatura

```
#define TEMP_IDEAL_MAX      15.0    // °C - Desenvolvimento f
úngico lento
#define TEMP_MODERADO_MIN  20.0    // °C - Início desenvolvi
mento fúngico médio
#define TEMP_MODERADO_MAX  30.0    // °C - Fim desenvolvimen
to fúngico médio
#define TEMP_CRITICO_MIN   40.0    // °C - Desenvolvimento f
úngico máximo
```

4.2 Thresholds de Umidade

```
#define HUMI_IDEAL_MAX      13.0    // % - Armazenamento seguro (≤13%)
#define HUMI_MODERADO_MIN  13.0    // % - Início crescimento fúngico rápid
o
#define HUMI_MODERADO_MAX  16.0    // % - Fim crescimento fúngico rápido
#define HUMI_CRITICO_MIN   16.0    // % - Crescimento fúngico "explosivo"
```

4.3 Fundamento Científico

Baseado na publicação "dicas-para-armazenamento-de-soja-2.pdf" da Embrapa Soja:

Parâmetro	Faixa	Impacto
Umidade	<13%	Armazenamento seguro prolongado
Umidade	13-16%	Crescimento fúngico rápido
Umidade	>16%	Crescimento fúngico explosivo
Temperatura	15°C	Desenvolvimento fúngico lento
Temperatura	40-55°C	Desenvolvimento fúngico máximo

Table 2: Parâmetros técnicos Embrapa para armazenagem de soja

5. ESTRUTURA DO CÓDIGO

5.1 Arquitetura de Software

// Estrutura principal
`setup()` → `loop()` → `lerSensor()` → `classificarCondicoes()` → `enviarThingSpeak()`

5.2 Variáveis Globais e Definições

```
// Configurações de amostragem
#define NUM_LEITURAS          6           // Amostras por ciclo
#define DELAY_ENTRE_LEITURAS 2000        // Intervalo entre leituras (ms)
#define SEND_INTERVAL         30000      // Envio para cloud (30s)

// Enumeração de status
typedef enum {
    STATUS_IDEAL = 1,
    STATUS_MODERADO = 2,
    STATUS_CRITICO = 3,
    STATUS_ERRO_SENSOR = 4
} StatusArmazenamento;
```

5.3 Campos ThingSpeak

```
#define THINGSPEAK_FIELD_TEMP      1    // Temperatura Atual (°C)
#define THINGSPEAK_FIELD_HUMI     2    // Umidade Atual (%)
#define THINGSPEAK_FIELD_STATUS   3    // Status (1-2-3-4)
#define THINGSPEAK_FIELD_HUMI_MIN 4    // Referência 13% (fixo)
```

6. SISTEMA DE CLASSIFICAÇÃO

6.1 Algoritmo de Classificação

```
StatusArmazenamento classificarCondicoes(float temperatura, float
umidade) {
    // 1. Verificação condições CRÍTICAS (prioridade máxima)
    if (temperatura >= 40.0 || umidade > 16.0)
        return STATUS_CRITICO;

    // 2. Verificação condições MODERADAS
    if ((temperatura >= 20.0 && temperatura <= 30.0) ||
        (umidade >= 13.0 && umidade <= 16.0))
        return STATUS_MODERADO;

    // 3. Condições IDEAIS
    return STATUS_IDEAL;
}
```

6.2 Matriz de Decisão

Condição	Temperatura	Umidade	Status	Ação
Ideal	≤15°C	≤13%	● 1	Manutenção
Moderado	20-30°C	13-16%	● 2	Vigilância
Crítico	≥40°C	≥16%	● 3	Intervenção
Erro	--	--	● 4	Manutenção

Table 3: Matriz de decisão de status

7. PROTOCOLOS DE COMUNICAÇÃO

7.1 Comunicação com DHT11

- Protocolo: Bus digital proprietário
- Timing: Requer delays precisos entre leituras
- Validação: Checksum interno do sensor
- Falhas: Retorna NaN em caso de erro

7.2 Comunicação Wi-Fi

```
void connectWiFi() {  
    WiFi.mode(WIFI_STA);  
    WiFi.begin(ssid, password);  
    // Timeout: 20 segundos (40 tentativas x 500ms)  
    // Fail-safe: Reinício automático em caso de falha  
}
```

7.3 API ThingSpeak

- Método: HTTP POST
- Frequência: 30 segundos
- Autenticação: API Key por channel
- Formato: Campos numéricos

8. GESTÃO DE ERROS

8.1 Sistema Fail-Safe

```
// 1. Validação de Leituras do sensor
if (!isnan(t) && t >= -10 && t <= 60) {
    // Leitura válida - processa
} else {
    // Leitura inválida - descarta e Loga
}

// 2. Critério de falha do sensor
if (leiturasInvalidas > (NUM_LEITURAS / 2)) {
    // Mais de 50% de Leituras inválidas
    return false; // Aborta ciclo
}

// 3. Recuperação de conexão Wi-Fi
if (WiFi.status() != WL_CONNECTED) {
    connectWiFi(); // Reconexão automática
}
```

8.2 Códigos de Erro ThingSpeak

```
void printThingSpeakStatus(int statusCode) {
    switch(statusCode) {
        case 200: Serial.println("OK - Sucesso");
        case 401: Serial.println("ERRO: API Key inválida");
        case 404: Serial.println("ERRO: Channel não encontrado");
        default: Serial.println("ERRO: Verificar conexão");
    }
}
```

8.3 Mecanismos de Recuperação

- Reconexão Wi-Fi automática
- Reinício do ESP32 em falhas graves
- Validação de múltiplas leituras
- Logs detalhados para diagnóstico

9. CONFIGURAÇÃO THINGSPEAK

9.1 Estrutura do Channel

Field	Nome	Tipo	Descrição
1	Temperatura	Número	Valor atual em °C
2	Umidade	Número	Valor atual em %
3	Status	Número	1=Normal, 2=Alerta, 3=Crítico, 4=Erro
4	Ref-Umidade	Número	Linha de referência 13%

Table 4: Estrutura dos campos do canal ThingSpeak

9.2 Configuração Recomendada dos Gráficos

Field 1 – Temperatura

```
{
  "Tipo": "Gráfico de Linha",
  "Eixo Y": "0 a 60°C",
  "Cores": "Azul",
  "Legenda": "Temperatura: {{value}}°C"
}
```

Field 2 – Umidade

```
{
  "Tipo": "Gráfico de Linha",
  "Eixo Y": "0 a 100%",
  "Cores": "Verde",
  "Legenda": "Umidade: {{value}}%"
}
```

Field 3 – Status

```
{
  "Tipo": "Gráfico Discreto",
  "Valores": [
    {"1": "Ideal", "cor": "#00FF00"},
    {"2": "Moderado", "cor": "#FFFF00"},
  ]
}
```

```
    {"3": "Crítico", "cor": "#FF0000"},  
    {"4": "Erro", "cor": "#808080"}  
  ]  
}
```

Field 4 - Referência

```
{  
  "Tipo": "Gráfico de Linha",  
  "Valor": "13.0",  
  "Cor": "Vermelho",  
  "Estilo": "Tracejado",  
  "Legenda": "Limite Seguro: 13%"  
}
```

10. FLUXO DE OPERAÇÃO

10.1 Sequência de Inicialização

Setup()

- Inicializa Serial Monitor
- Conecta Wi-Fi
- Inicializa sensor DHT11 (GPIO4)
- Configura cliente ThingSpeak

Loop() Principal

- Verifica intervalo de envio (30s)
- Valida conexão Wi-Fi
- Executa ciclo de leitura do sensor
- Classifica condições
- Transmite dados para cloud
- Aguarda próximo ciclo

10.2 Ciclo de Leitura do Sensor

```
for (int i = 0; i < NUM_LEITURAS; i++) {  
    // Leitura temperatura e umidade  
    // Validação de faixa  
    // Acumulação para média  
    // Delay de estabilização  
}  
// Cálculo da média  
// Retorno sucesso/falha
```

11.CONSIDERAÇÕES TÉCNICAS

11.1 Precisão e Estabilidade

- Múltiplas leituras: 6 amostras por ciclo reduzem ruído
- Validação de faixa: Descarta valores fisicamente impossíveis
- Média móvel: Suaviza variações abruptas

11.2 Consumo e Performance

- **Wi-Fi:** Conecta apenas durante transmissão
- **CPU:** Baixo consumo entre ciclos
- **Memória:** Otimizada para ESP32-C3

11.3 Manutenção e Atualizações

- **Logs detalhados:** Facilitam diagnóstico remoto
- **Reinício automático:** Recuperação de falhas graves
- **Configuração flexível:** Parâmetros via defines

11.4 Manutenção Preventiva

- Verificação mensal: Calibração do sensor DHT11
- Verificação trimestral: Conexões físicas e alimentação
- Atualização semestral: Firmware e bibliotecas

12. PROCEDIMENTOS DE EMERGÊNCIA

12.1 Condições Críticas Detectadas

Condição	Threshold	Ação Imediata
Temperatura Alta	≥40°C	Acionar sistema de resfriamento
Umidade Elevada	>16%	Ativar sistema de secagem
Falha de Sensor	Status 4	Verificação física do equipamento

Table 5: Matriz de emergências e ações

12.2 Resposta do Sistema

- **Status 3 (Crítico):** Alertas visuais imediatos no dashboard
- **Status 4 (Erro):** Notificação para manutenção técnica
- **Reinício automático:** Em falhas de conexão persistentes tentativas

ANEXOS

A. Código Fonte Completo

Disponível em: `agrosilo_esp32c3.ino`

B. Configuração ThingSpeak

- Channel ID: [Inserir ID]
- API Key: [Inserir chave]
- URL: [https://thingspeak.com/channels/\[ID\]](https://thingspeak.com/channels/[ID])

C. Referências Técnicas

- Embrapa Soja: "dicas-para-armazenamento-de-soja-2.pdf"
 - <https://www.infoteca.cnptia.embrapa.br/infoteca/bitstream/doc/1155426/1/Circ-Tec-196.pdf>
 - Documentação DHT11: Datasheet
 - ESP32-C3: Technical Reference Manual
 - https://documentation.espressif.com/esp32-c3_technical_reference_manual_en.pdf
-

Documento técnico atualizado em conformidade com as especificações Embrapa Soja - Versão 2.6

Sistema desenvolvido para garantir a qualidade do armazenamento de soja e prevenir perdas por fatores ambientais.