

Documentação Técnica - Controlador Python

1. Introdução ao Firmware ESP32

O firmware do ESP32 é responsável por executar o controle físico de uma esteira ergométrica com aquecimento e modulação de pressão, além do controle de iluminação LED e periféricos como aromatizador e neon.

A aplicação roda em tempo real e utiliza múltiplas interrupções, tarefas FreeRTOS e timers de hardware para garantir precisão e resposta rápida.

2. Sensores e Atuadores

[Sensor] Sensores e Atuadores:

- Sensor de Temperatura e Pressão: BME280
- Display LCD I2C: Mostra temperatura, pressão, setpoints e status
- Controle de potência via TRIAC: Motor (bomba) e aquecedor
- LEDs WS2812: controlados com FastLED
- Relés: Lâmpada, esteira
- GPIOs: Aromatizador, Neon
- Botão de Emergência: Aciona desligamento imediato

Pinos de uso:

- Motor: GPIO 25
- Aquecedor: GPIO 26
- Zero-cross: GPIO 27
- Botão emergência: GPIO 5
- Relé lâmpada: GPIO 14
- Relé esteira: GPIO 32
- LED neon: GPIO 33
- Aromatizador: GPIO 4

3. Modos de Operação

Documentação Técnica - Controlador Python

[Modo] Modos de Operação:

- **Automático**: PID calcula a potência do aquecedor e motor com base nos setpoints definidos pelo controlador.
- **Manual**: Potência de bomba e aquecedor é controlada diretamente via comandos recebidos do Python.
- Alternância entre modos é feita com o comando ID_MODAL.

4. Controle PID

[PID] Controle PID:

- Temperatura:
 - $K_p = 90.0$
 - $K_i = 35.0$
 - $K_d = 30.0$
- Pressão:
 - $K_p = 2.0$
 - $K_i = 0.5$
 - $K_d = 1.0$
- Intervalo de controle ajustado a cada laço do loop principal
- Output limitado de 5000 a 20000 (convertido para 0-100% para controle AC)

5. Comunicação Serial

[Serial] Comunicação Serial:

- Serial principal: comunicação com o Python (115200 bps)
- Serial secundária: comunicação com a esteira (9600 bps, SERIAL_8E1)
- Protocolo de pacote:

[0x02][ID_CHIP][ID_COMANDO][VALORES...][0x03]

Exemplo de recebimento:

- [0x02, 0x08, 0x13, Rext, Gext, Bext, Rint, Gint, Bint, 0x03] = Atualiza cores RGB

Documentação Técnica - Controlador Python

Exemplo de resposta:

- [0x02, 0x08, 0x0B, byte1, byte2, 0x03] = Temperatura atual

6. Funções e Processamento

[Funções] Funções e Comportamentos:

- ``processPCIncomingByte()``: monta pacotes byte a byte recebidos do Python
- ``process_PC_data()``: interpreta e executa comandos
- ``lerSensores()``: lê o BME280 e atualiza variáveis globais
- ``ajustarPotencia()``: executa o PID e aplica potência via timers
- ``updateLCD()``: mostra temperatura, pressão, setpoints e status
- ``enviaComandosEsteiraTask()``: envia velocidade/inclinação periodicamente
- ``handleEmergency()``: tratamento de parada de emergência (botão físico)
- ``processarComandoRGB()``: aplica cor aos LEDs WS2812

LEDs são atualizados com FastLED e brilho limitado a 20%. Estado dos LEDs pode ser salvo/restaurado.