
AIoT_Sentinel 4.0 – Código Core do Arduino

Responsável por interpretar comandos recebidos do Raspberry Pi via Serial e acionar respostas físicas (LEDs e Buzzer) com base na autorização.

```
*/

#define BUZZER_PIN 8

#define LED_VERDE 9

#define LED_VERMELHO 10

String comandoSerial = "";

bool sistemaAtivo = true;

void setup() {

    Serial.begin(9600);

    pinMode(BUZZER_PIN, OUTPUT);

    pinMode(LED_VERDE, OUTPUT);

    pinMode(LED_VERMELHO, OUTPUT);

    Serial.println("Sentinel Arduino iniciado. Aguardando comandos do núcleo inteligente...");

    standbyFeedback();

}

void loop() {

    if (Serial.available() > 0) {

        comandoSerial = Serial.readStringUntil('\n');
```

```
comandoSerial.trim();

if (comandoSerial.startsWith("AUTORIZAR:")) {

    String nome = comandoSerial.substring(10);

    acessoAutorizado(nome);

} else if (comandoSerial.startsWith("NEGAR:")) {

    String nome = comandoSerial.substring(6);

    acessoNegado(nome);

} else if (comandoSerial == "SHUTDOWN") {

    sistemaAtivo = false;

    desligarSistema();

} else if (comandoSerial == "BOOT") {

    sistemaAtivo = true;

    Serial.println("Sistema reativado.");

    standbyFeedback();

} else {

    Serial.println("Comando desconhecido.");

}

}

}

void acessoAutorizado(String nome) {

    if (!sistemaAtivo) {

        Serial.println("Sistema inativo. Use BOOT para reativar.");

    }

}
```

```

    return;
}

Serial.println("Acesso autorizado: " + nome);

digitalWrite(LED_VERDE, HIGH);

digitalWrite(LED_VERMELHO, LOW);

tone(BUZZER_PIN, 1000, 200); // Beep curto

delay(1000);

standbyFeedback();
}

void acessoNegado(String nome) {

    if (!sistemaAtivo) {

        Serial.println("Sistema inativo. Use BOOT para reativar.");

        return;
    }

    Serial.println("Acesso negado: " + nome);

    digitalWrite(LED_VERDE, LOW);

    digitalWrite(LED_VERMELHO, HIGH);

    tone(BUZZER_PIN, 400, 800); // Alarme longo

    delay(2000);

    standbyFeedback();
}

void desligarSistema() {

    Serial.println("Sistema Sentinel desligado.");

    digitalWrite(LED_VERDE, LOW);

```

```
digitalWrite(LED_VERMELHO, LOW);

noTone(BUZZER_PIN);
}

void standbyFeedback() {

digitalWrite(LED_VERDE, LOW);

digitalWrite(LED_VERMELHO, LOW);

noTone(BUZZER_PIN);
}
```