

Nome: Ingrid Alves Oliveira de Jesus

BI3018521

Nome: Mariana Sales de Oliveira Lazarini

BI3017206

Microcontroladores

Projeto a ser implementado: Gerenciamento de Semáforo

Plataforma alvo: ESP 32

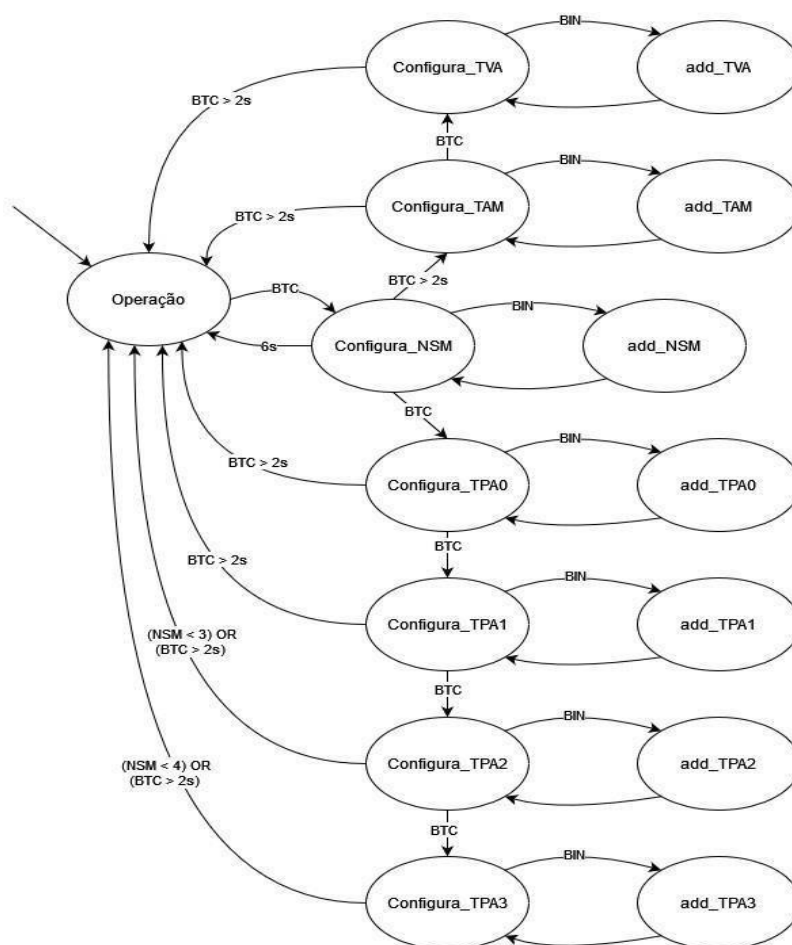
Mapeamento físico das I/Os:

Nome	Pino
BTC	17
BIN	2
SSP0	12
SSP1	13
SSP2	4
SSP3	0
LAM0	21
LVM0	15
LVD0	22
LAM1	33
LVM1	25
LVD1	32
LAM2	27
LVM2	14
LVD2	26
LAM3	18

LVM3	5
LVD3	19
LED_C	16

Codificação do(s) modelo(s) lógico(s):

Figura 3 - Diagrama de estados do modo de ajuste de parâmetros.



Código do diagrama:

```
#include <WiFi.h>
```

```
#include <PubSubClient.h>
```

```
const char* ssid = "Wokwi-GUEST";
```

```
const char* key = "";
```

```
const char* broker = "test.mosquitto.org";
```

```
int port = 1883;
```

```
WiFiClient wifiClient;
```

```
PubSubClient mqttClient(wifiClient);
```

```
byte AtivarSemaforo;
```

```
void conexaoWiFi() {  
    //Conexão ao Wi-Fi  
    Serial.print("Conectando-se ao Wi-Fi ");  
    Serial.print(ssid);  
    Serial.print(" ");  
    WiFi.begin(ssid, key, 6);  
    while (WiFi.status() != WL_CONNECTED) {  
        delay(100);  
        Serial.print(".");  
    }  
    Serial.println(" Conectado!");  
}
```

```
void conexaoBroker() {  
    //Conexão ao Broker  
    mqttClient.setServer(broker, port);  
    while (!mqttClient.connected()) {  
        Serial.print("Conectando-se ao broker ");  
        //if (mqttClient.connect(WiFi.macAddress().c_str())) {  
        if (mqttClient.connect("87987ji0938j1289KJSAUE3")) {  
            Serial.println(" Conectado!");  
        } else {  
            Serial.print(".");  
            delay(100);  
        }  
    }  
}
```

```
#define BTC 17 // Botão de configuração
```

```
#define BIN 2 // Botão de incremento
```

```
#define SSP0 12
```

```
#define SSP1 13
```

```
#define SSP2 4
```

```
#define SSP3 0
```

```
// Definição dos pinos dos LEDs
```

```
#define LAM0 21 // Amarelo (Sensor 1)
```

```
#define LVM0 15 // Vermelho (Sensor 1)
```

```
#define LVD0 22 // Verde (Sensor 1)
```

```
#define LAM1 33 // Amarelo (Sensor 2)
```

```
#define LVM1 25 // Vermelho (Sensor 2)
```

```
#define LVD1 32 // Verde (Sensor 2)
```

```
#define LAM2 27 // Amarelo (Sensor 3)
```

```
#define LVM2 14 // Vermelho (Sensor 3)
```

```
#define LVD2 26 // Verde (Sensor 3)
```

```
#define LAM3 18 // Amarelo (Sensor 4)
```

```
#define LVM3 5 // Vermelho (Sensor 4)
```

```
#define LVD3 19 // Verde (Sensor 4)
```

```
#define LED_C 16 // Definição do LED de modo de configuração
```

```
// Declaração de variáveis globais, com o tempo.
```

```
int NSM; // Número de semáforos (2 a 4, padrão: 2)
```

```
int TAM; // Tempo da lâmpada amarela (5 a 10 segundos, padrão: 5)
```

```
int TVA; // Tempo vermelho adicional (3 a 10 segundos, padrão: 3)
```

```
int TPA0; // Tempo de abertura do semáforo 0 (15 a 60 segundos, padrão: 15)
```

```
int TPA1; // Tempo de abertura do semáforo 1
```

```
int TPA2; // Tempo de abertura do semáforo 2
```

```
int TPA3; // Tempo de abertura do semáforo 3
```

```
unsigned long IniTemPar; //Tempozização do modo de configuração
unsigned long FimTemPar; //Tempozização do modo de configuração
unsigned long IniTemSem; //Tempozização dos semáforos
unsigned long FimTemSem; //Tempozização dos semáforos
unsigned long IniTemS0; //Tempozização do semáforo 0
unsigned long FimTemS0; //Tempozização do semáforo 0
unsigned long IniTemS1; //Tempozização do semáforo 1
unsigned long FimTemS1; //Tempozização do semáforo 1
unsigned long IniTemS2; //Tempozização do semáforo 2
unsigned long FimTemS2; //Tempozização do semáforo 2
unsigned long IniTemS3; //Tempozização do semáforo 3
unsigned long FimTemS3; //Tempozização do semáforo 3
```

```
// Definição dos enums
```

```
enum Semaforo0 {
    Verde0 = 1,
    Amarelo0 = 2,
    Vermelho0 = 3
} S0;
```

```
enum Semaforo1 {
    Verde1 = 1,
    Amarelo1 = 2,
    Vermelho1 = 3
} S1;
```

```
enum Semaforo2 {
    Verde2 = 1,
    Amarelo2 = 2,
    Vermelho2 = 3
} S2;
```

```
enum Semaforo3 {
    Verde3 = 1,
```

```
    Amarelo3 = 2,  
    Vermelho3 = 3  
} S3;
```

```
enum AjusteParametro {  
    Modo_de_operacao = 0,  
    Configura_NSM = 1,  
    add_NSM = 2,  
    Configura_TAM = 3,  
    add_TAM = 4,  
    Configura_TVA = 5,  
    add_TVA = 6,  
    Configura_TPA0 = 7,  
    add_TPA0 = 8,  
    Configura_TPA1 = 9,  
    add_TPA1 = 10,  
    Configura_TPA2 = 11,  
    add_TPA2 = 12,  
    Configura_TPA3 = 13,  
    add_TPA3 = 14  
} AP;
```

```
//AjusteParametro AP = Modo_de_operacao;
```

```
enum AberturaSemafaro {  
    SMF0 = 0,  
    SMF1 = 1,  
    SMF2 = 2,  
    SMF3 = 3  
} AS;
```

```
//Processamentos configuração  
void Processamento_Modo_de_operacao(){  
    digitalWrite(LED_C, LOW);
```

```
if(digitalRead(BTC)){  
    delay(300);  
    AP = Configura_NSM;  
    IniTemPar = millis();  
}  
}
```

```
void Processamento_Configura_NSM(){  
    FimTemPar = millis();  
    digitalWrite(LED_C, HIGH);  
    if(FimTemPar-IniTemPar>=6000){  
        AP = Modo_de_operacao;  
    }  
    else if(digitalRead(BTC)){  
        delay(300);  
        AP = Configura_TAM;  
    }  
    else if(digitalRead(BIN)){  
        AP = add_NSM;  
    }  
}
```

```
void Processamento_add_NSM(){  
    digitalWrite(LED_C, HIGH);  
    if(NSM == 4){  
        NSM = 2;  
    }  
    else{  
        NSM++;  
    }  
    AP = Configura_NSM;  
    IniTemPar = millis();  
}
```

```

void Processamento_Configura_TAM(){
    digitalWrite(LED_C, HIGH);
    if(digitalRead(BTC)){
        IniTemPar = millis();
        while(digitalRead(BTC)){
            FimTemPar = millis();
        }
        if(FimTemPar-IniTemPar> 2000){
            AP = Modo_de_operacao;
        }
        else{
            AP = Configura_TVA;
        }
    }
    else if(digitalRead(BIN)){
        AP = add_TAM;
    }
}

```

```

void Processamento_add_TAM(){
    digitalWrite(LED_C, HIGH);
    if(TAM == 10){
        TAM = 5;
    }
    else{
        TAM++;
    }
    AP = Configura_TAM;
}

```

```

void Processamento_Configura_TVA(){
    digitalWrite(LED_C, HIGH);
    if(digitalRead(BTC)){

```



```

IniTemPar = millis();
while(digitalRead(BTC)){
    FimTemPar = millis();
}
if(FimTemPar-IniTemPar> 2000){
    AP = Modo_de_operacao;
}
else{
    AP = Configura_TPA0;
}
}
else if(digitalRead(BIN)){
    AP = add_TVA;
}
}

```

```

void Processamento_add_TVA(){
    digitalWrite(LED_C, HIGH);
    if(TVA == 10){
        TVA = 3;
    }
    else{
        TVA++;
    }
    AP = Configura_TVA;
}

```

```

void Processamento_Configura_TPA0(){
    digitalWrite(LED_C, HIGH);
    if(digitalRead(BTC)){
        IniTemPar = millis();
        while(digitalRead(BTC)){
            FimTemPar = millis();

```

```

    }

    if(FimTemPar-IniTemPar> 2000){

        AP = Modo_de_operacao;

    }

    else{

        AP = Configura_TPA1;

    }

}

else if(digitalRead(BIN)){

    AP = add_TPA0;

}

}

```

```

void Processamento_add_TPA0(){

    digitalWrite(LED_C, HIGH);

    if(TPA0 == 60){

        TPA0 = 15;

    }

    else{

        TPA0 += 5;

    }

    AP = Configura_TPA0;

}

```

```

void Processamento_Configura_TPA1(){

    digitalWrite(LED_C, HIGH);

    if(digitalRead(BTC)){

        IniTemPar = millis();

        while(digitalRead(BTC)){

            FimTemPar = millis();

        }

        if(FimTemPar-IniTemPar> 2000){

            AP = Modo_de_operacao;

        }

    }

}

```

```

else{
    AP = Configura_TPA2;
}
}
else if(digitalRead(BIN)){
    AP = add_TPA1;
}
}

```

```

void Processamento_add_TPA1(){
    digitalWrite(LED_C, HIGH);
    if(TPA1 == 60){
        TPA1 = 15;
    }
    else{
        TPA1 += 5;
    }
    AP = Configura_TPA1;
}

```

```

void Processamento_Configura_TPA2(){
    digitalWrite(LED_C, HIGH);
    if(NSM < 3){
        AP = Modo_de_operacao;
        IniTemPar = millis();
    }
    else if(digitalRead(BTC)){
        IniTemPar = millis();
        while(digitalRead(BTC)){
            FimTemPar = millis();
        }
        if(FimTemPar-IniTemPar> 2000){
            AP = Modo_de_operacao;
        }
    }
}

```

```

else{
    IniTemPar = millis();
    AP = Configura_TPA3;
}
}
else if(digitalRead(BIN)){
    AP = add_TPA2;
    IniTemPar = millis();
}
}

```

```

void Processamento_add_TPA2(){
    digitalWrite(LED_C, HIGH);
    if(TPA2 == 60){
        TPA2 = 15;
    }
    else{
        TPA2 += 5;
    }
    AP = Configura_TPA2;
}

```

```

void Processamento_Configura_TPA3(){
    digitalWrite(LED_C, HIGH);
    FimTemPar = millis();
    if(FimTemPar-IniTemPar > 6000 || NSM < 4){
        AP = Modo_de_operacao;
    }
    else if(digitalRead(BIN)){
        AP = add_TPA3;
    }
}

```

```

void Processamento_add_TPA3(){

```

```

digitalWrite(LED_C, HIGH);
if(TPA3 == 60){
    TPA3 = 15;
}
else{
    TPA3 += 5;
}
IniTemPar = millis();
AP = Configura_TPA3;
}

//Processamentos abertura dos semáforos
void Processamento_SMF0(){
    FimTemSem = millis();
    if(AtivarSemaforo == 1){
        mqttClient.publish("EstadoSem","Semáforo 0 Abriu !!");
        S0 = Verde0;
        AtivarSemaforo = 0;
    }
    if((FimTemSem - IniTemSem >= (TPA0+TAM+TVA)*1000) && (digitalRead(SSP1))){
        IniTemS1 = millis();
        IniTemSem = millis();
        AS = SMF1;
        AtivarSemaforo = 1;
    }
    else if((FimTemSem - IniTemSem >= (TPA0+TAM+TVA)*1000) && (!digitalRead(SSP1) &&
digitalRead(SSP2))){
        IniTemS2 = millis();
        IniTemSem = millis();
        AS = SMF2;
        AtivarSemaforo = 1;
    }
    else if((FimTemSem - IniTemSem >= (TPA0+TAM+TVA)*1000) && (!digitalRead(SSP1) &&
!digitalRead(SSP2) && digitalRead(SSP3))){
        IniTemS3 = millis();

```

```

    IniTemSem = millis();
    AS = SMF3;
    AtivarSemaforo = 1;
}
}

void Processamento_SMF1(){
    FimTemSem = millis();
    if(AtivarSemaforo == 1){
        mqttClient.publish("EstadoSem", "Semáforo 1 Abriu !!");
        S1 = Verde1;
        AtivarSemaforo = 0;
    }
    if((FimTemSem - IniTemSem >= (TPA1+TAM+TVA)*1000) && (digitalRead(SSP2))){
        IniTemS2 = millis();
        IniTemSem = millis();
        AS = SMF2;
        AtivarSemaforo = 1;
    }
    else if((FimTemSem - IniTemSem >= (TPA1+TAM+TVA)*1000) && (!digitalRead(SSP2) &&
digitalRead(SSP3))){
        IniTemS3 = millis();
        IniTemSem = millis();
        AS = SMF3;
        AtivarSemaforo = 1;
    }
    else if((FimTemSem - IniTemSem >= (TPA1+TAM+TVA)*1000) && (!digitalRead(SSP2) &&
!digitalRead(SSP3) && digitalRead(SSP0))){
        IniTemS0 = millis();
        IniTemSem = millis();
        AS = SMF0;
        AtivarSemaforo = 1;
    }
}
}

```

```

void Processamento_SMF2(){
    FimTemSem = millis();
    if(AtivarSemaforo == 1){
        mqttClient.publish("EstadoSem","Semáforo 2 Abriu !!");
        S2 = Verde2;
        AtivarSemaforo = 0;
    }
    if((FimTemSem - IniTemSem >= (TPA2+TAM+TVA)*1000) && (digitalRead(SSP3))){
        IniTemS3 = millis();
        IniTemSem = millis();
        AS = SMF3;
        AtivarSemaforo = 1;
    }
    else if((NSM < 3) || ((FimTemSem - IniTemSem >= (TPA2+TAM+TVA)*1000) && (!digitalRead(SSP3) && digitalRead(SSP0)))){
        IniTemS0 = millis();
        IniTemSem = millis();
        AS = SMF0;
        AtivarSemaforo = 1;
    }
    else if((FimTemSem - IniTemSem >= (TPA2+TAM+TVA)*1000) && (!digitalRead(SSP3) && !digitalRead(SSP0) && digitalRead(SSP1))){
        IniTemS1 = millis();
        IniTemSem = millis();
        AS = SMF1;
        AtivarSemaforo = 1;
    }
}

```

```

void Processamento_SMF3(){
    FimTemSem = millis();
    if(AtivarSemaforo == 1){
        mqttClient.publish("EstadoSem","Semáforo 3 Abriu !!");
        S3 = Verde3;
        AtivarSemaforo = 0;
    }
}

```

```

}

if((NSM < 4) || ((FimTemSem - IniTemSem >= (TPA3+TAM+TVA)*1000) &&
(digitalRead(SSP0)))){

    IniTemS0 = millis();

    IniTemSem = millis();

    AS = SMF0;

    AtivarSemaforo = 1;

}

else if((FimTemSem - IniTemSem >= (TPA3+TAM+TVA)*1000) && (!digitalRead(SSP0) &&
digitalRead(SSP1))){

    IniTemS1 = millis();

    IniTemSem = millis();

    AS = SMF1;

    AtivarSemaforo = 1;

}

else if((FimTemSem - IniTemSem >= (TPA3+TAM+TVA)*1000) && (!digitalRead(SSP0) &&
!digitalRead(SSP1) && digitalRead(SSP2))){

    IniTemS2 = millis();

    IniTemSem = millis();

    AS = SMF2;

    AtivarSemaforo = 1;

}

}

//Processamentos Semáforo 0

void ProcessamentoVerde0(){

    digitalWrite(LVD0, HIGH);

    digitalWrite(LAM0, LOW);

    digitalWrite(LVM0, LOW);

    FimTemS0 = millis();

    if((FimTemS0 >= TPA0*1000) && (digitalRead(SSP1) || digitalRead(SSP2) || digitalRead(SSP3))){

        S0 = Amarelo0;

        IniTemS0 = millis();

    }

}

```



```

void ProcessamentoAmarelo0(){
    digitalWrite(LVD0, LOW);
    digitalWrite(LAM0, HIGH);
    digitalWrite(LVM0, LOW);
    FimTemS0 = millis();
    if(FimTemS0 >= TAM*1000){
        S0 = Vermelho0;
    }
}

```

```

void ProcessamentoVermelho0(){
    digitalWrite(LVD0, LOW);
    digitalWrite(LAM0, LOW);
    digitalWrite(LVM0, HIGH);
}

```

//Processamentos Semáforo 1

```

void ProcessamentoVerde1(){
    digitalWrite(LVD1, HIGH);
    digitalWrite(LAM1, LOW);
    digitalWrite(LVM1, LOW);
    FimTemS1 = millis();
    if((FimTemS1 >= TPA1*1000) && (digitalRead(SSP0) || digitalRead(SSP2) || digitalRead(SSP3))){
        S1 = Amarelo1;
        IniTemS1 = millis();
    }
}

```

```

void ProcessamentoAmarelo1(){
    digitalWrite(LVD1, LOW);
    digitalWrite(LAM1, HIGH);
    digitalWrite(LVM1, LOW);
    FimTemS1 = millis();
}

```

```
if(FimTemS1 >= TAM*1000){  
    S1 = Vermelho1;  
}  
}
```

```
void ProcessamentoVermelho1(){  
    digitalWrite(LVD1, LOW);  
    digitalWrite(LAM1, LOW);  
    digitalWrite(LVM1, HIGH);  
}
```

//Processamentos Semáforo 2

```
void ProcessamentoVerde2(){  
    digitalWrite(LVD2, HIGH);  
    digitalWrite(LAM2, LOW);  
    digitalWrite(LVM2, LOW);  
    FimTemS2 = millis();  
    if((FimTemS2 >= TPA2*1000) && (digitalRead(SSP0) || digitalRead(SSP1) || digitalRead(SSP3))){  
        S2 = Amarelo2;  
        IniTemS2 = millis();  
    }  
}
```

```
void ProcessamentoAmarelo2(){  
    digitalWrite(LVD2, LOW);  
    digitalWrite(LAM2, HIGH);  
    digitalWrite(LVM2, LOW);  
    FimTemS2 = millis();  
    if(FimTemS2 >= TAM*1000){  
        S2 = Vermelho2;  
    }  
}
```

```
void ProcessamentoVermelho2(){
```

```
digitalWrite(LVD2, LOW);  
digitalWrite(LAM2, LOW);  
digitalWrite(LVM2, HIGH);  
}
```

```
//Processamentos Semáforo 3
```

```
void ProcessamentoVerde3(){  
    digitalWrite(LVD3, HIGH);  
    digitalWrite(LAM3, LOW);  
    digitalWrite(LVM3, LOW);  
    FimTemS3 = millis();  
    if((FimTemS3 >= TPA3*1000) && (digitalRead(SSP0) || digitalRead(SSP1) || digitalRead(SSP2))){  
        S3 = Amarelo3;  
        IniTemS3 = millis();  
    }  
}
```

```
void ProcessamentoAmarelo3(){  
    digitalWrite(LVD3, LOW);  
    digitalWrite(LAM3, HIGH);  
    digitalWrite(LVM3, LOW);  
    FimTemS3 = millis();  
    if(FimTemS3 >= TAM*1000){  
        S3 = Vermelho3;  
    }  
}
```

```
void ProcessamentoVermelho3(){  
    digitalWrite(LVD3, LOW);  
    digitalWrite(LAM3, LOW);  
    digitalWrite(LVM3, HIGH);  
}
```

```
void setup() {  
    // Inicializa comunicação serial  
  
    // Configuração dos pinos dos sensores PIR  
    pinMode(SSP0, INPUT);  
    pinMode(SSP1, INPUT);  
    pinMode(SSP2, INPUT);  
    pinMode(SSP3, INPUT);  
  
    // Configuração dos LEDs dos sensores  
    pinMode(LAM0, OUTPUT);  
    pinMode(LVM0, OUTPUT);  
    pinMode(LVD0, OUTPUT);  
    pinMode(LAM1, OUTPUT);  
    pinMode(LVM1, OUTPUT);  
    pinMode(LVD1, OUTPUT);  
    pinMode(LAM2, OUTPUT);  
    pinMode(LVM2, OUTPUT);  
    pinMode(LVD2, OUTPUT);  
    pinMode(LAM3, OUTPUT);  
    pinMode(LVM3, OUTPUT);  
    pinMode(LVD3, OUTPUT);  
  
    // Configuração dos LEDs gerais e do botão  
    pinMode(LED_C, OUTPUT);  
    pinMode(BTC, INPUT);  
    pinMode(BIN, INPUT);  
  
    //Inicialização das Máquinas de Estados (AP e AS)  
    AP = Modo_de_operacao; // Inicializa máquina de ajuste de parâmetros  
    AS = SMF0;             // Inicializa máquina de controle de semáforo
```

```

// Inicialização de variáveis globais
NSM = 4; // Número de semáforos
TAM = 15; // Tempo da lâmpada amarela
TVA = 3; // Tempo vermelho adicional
TPA0 = 15; // Tempo de abertura do semáforo 0
TPA1 = 15;
TPA2 = 15;
TPA3 = 15;

// Inicialização dos estados do semáforo
S0 = Vermelho0;
S1 = Vermelho1;
S2 = Vermelho2;
S3 = Vermelho3;

AtivarSemaforo = 1;

// Exibe mensagem de inicialização
Serial.begin(115200);
Serial.println("Sistema iniciado. Monitorando sensores PIR...");

//Inicialização da contagem de tempo do semáforo
IniTemSem = millis();
IniTemS0 = millis();
conexaoWiFi();
conexaoBroker();
}

void loop() {
// Máquina de controle dos semáforos
switch (AS) {
case SMF0: Processamento_SMF0();
break;

```

```
case SMF1: Processamento_SMF1();  
    break;
```

```
case SMF2: Processamento_SMF2();  
    break;
```

```
case SMF3: Processamento_SMF3();  
    break;  
}
```

```
switch (S0){  
    case Verde0: ProcessamentoVerde0();  
        break;  
    case Amarelo0: ProcessamentoAmarelo0();  
        break;  
    case Vermelho0: ProcessamentoVermelho0();  
        break;  
}
```

```
switch (S1){  
    case Verde1: ProcessamentoVerde1();  
        break;  
    case Amarelo1: ProcessamentoAmarelo1();  
        break;  
    case Vermelho1: ProcessamentoVermelho1();  
        break;  
}
```

```
switch (S2){  
    case Verde2: ProcessamentoVerde2();  
        break;  
    case Amarelo2: ProcessamentoAmarelo2();  
        break;  
    case Vermelho2: ProcessamentoVermelho2();
```

```
    break;
}
```

```
switch (S3){
    case Verde3: ProcessamentoVerde3();
        break;
    case Amarelo3: ProcessamentoAmarelo3();
        break;
    case Vermelho3: ProcessamentoVermelho3();
        break;
}
```

```
switch (AP) {
    case Modo_de_operacao: Processamento_Modo_de_operacao();
        break;
    case Configura_NSM: Processamento_Configura_NSM();
        break;
    case add_NSM: Processamento_add_NSM();
        break;
    case Configura_TAM: Processamento_Configura_TAM();
        break;
    case add_TAM: Processamento_add_TAM();
        break;
    case Configura_TVA: Processamento_Configura_TVA();
        break;
    case add_TVA: Processamento_add_TVA(); break;
    case Configura_TPA0: Processamento_Configura_TPA0();
        break;
    case add_TPA0: Processamento_add_TPA0();
        break;
    case Configura_TPA1: Processamento_Configura_TPA1();
        break;
    case add_TPA1: Processamento_add_TPA1();
        break;
}
```

```

    case Configura_TPA2: Processamento_Configura_TPA2();
        break;
    case add_TPA2: Processamento_add_TPA2();
        break;
    case Configura_TPA3: Processamento_Configura_TPA3();
        break;
    case add_TPA3: Processamento_add_TPA3();
        break;
}
Serial.print("TEMP: ");
Serial.print((FimTemSem-IniTemSem));
Serial.print(" || ");
Serial.println(AS);
/*
Serial.print("Parametro: ");
Serial.println(AP);
Serial.print("NSM: ");
Serial.print(NSM);
Serial.print(" || TAM: ");
Serial.print(TAM);
Serial.print(" || TVA: ");
Serial.print(TVA);
Serial.print(" || TPA0: ");
Serial.print(TPA0);
Serial.print(" || TPA1: ");
Serial.print(TPA1);
Serial.print(" || TPA2: ");
Serial.print(TPA2);
Serial.print(" || TPA3: ");
Serial.println(TPA3);
*/
delay(200);
}

```


Link da simulação no Wokwi : <https://wokwi.com/projects/421264975196144641>