IntelligentDestiller

Generated by Doxygen 1.12.0

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

Chapter 2

File Documentation

2.1 C:/workspace/Projeto/InteligentDestiller/include/AP.h File Reference

Header file for the AP project.

```
#include "MD.h"
```

Include dependency graph for AP.h:

2.2 AP.h

Go to the documentation of this file.

```


#ifndef _AP_h
#define _AP_h

#include "MD.h"

void destiler();
void setupRoutes();

#endif
```

2.3 C:/workspace/Projeto/InteligentDestiller/include/MD.h File Reference

Header file for the MD project.

```
#include "OS.h"
```

Include dependency graph for MD.h: This graph shows which files directly or indirectly include this file:

Macros

- #define ALARM_TIME_ON 600

  Processor.
- #define ALARM_TIME_OFF 200
- #define WIFI_MODE_OPTIONS 3

  Testa as entradas e saidas.

Functions

- void toggleAutoMode ()

  MiddleWare Header.
- void toggleIndAuto ()

  Toggle auto mode.
- void indicatorsManagement ()

  Toggle auto indicator.
- void modeManagement ()

  Indicators management.
- void updateOutputsWeb ()

  Mode management.
- void toggleAutoModeWeb ()

  Update outputs for the web interface.
- void togglePumpWeb ()

  Toggle auto mode.
- void toggleValveWaterInWeb ()

  Toggle pump state.
- void toggleValveWaterOutWeb ()

  Toggle valve water in.
- void toggleResistorWeb ()

  Toggle valve water out.
- void setAutoModeWeb (bool state)

  Toggle resistor.
- void setPumpWeb (bool state)

  Set auto mode.
- void setValveWaterInWeb (bool state)

  Set pump state.
- void setValveWaterOutWeb (bool state)

  Set valve water in.
- void setResistorWeb (bool state)

  Set valve water out.
- bool getAutoModeWeb ()

  Set resistor.
- bool getPumpWeb ()

  Get auto mode.
- bool getValv_Water_InWeb ()

  Get pump state.
- bool getValv_Water_OutWeb ()

  Get valve water in.
- bool getResistorWeb ()

  Get valve water out.
- String handleTimer ()

  Get resistor.
- String handleHour ()

  Handle timer.
- String handleMinute ()

  Handle hour.
- String handleSecound ()

  Handle minute.
- String handleWifiQuality ()

Handle secound.

- String handleWifiSSID ()

  Handle wifi quality.
- String handleWifiTX ()

  Handle wifi SSID.
- String handleWifiIP ()

  Handle wifi TX.
- String handleTimeAdjust (const String &adjustment)

  Handle wifi IP.
- String handleTimerControl (const String &control)

  Handle time adjustment.
- String handleState (const String &state)

  Handle timer control.
- String handleToggle (const String &toggle)

  Handle state.
- String processor (const String &var)

  Handle toggle.
- long webTimer (String op, int16_t amount)

  Timer Functions calls.
- int getTimerSecound ()

  Timer for web operations.
- int getTimerMinute ()

  Get timer secound.
- int getTimerHour ()

  Get timer minute.
- bool getTimerStatus ()

  Get timer hour.
- void add1s ()

  Get timer status.
- void add5s ()

  Add 1 secound to timer.
- void add10s ()

  Add 5 secounds to timer.
- void add1m ()

  Add 10 secounds to timer.
- void add5m ()

  Add 1 minute to timer.
- void add10m ()

  Add 5 minutes to timer.
- void rem1s ()

  Add 10 minutes to timer.
- void rem5s ()

  Remove 1 secound from timer.
- void rem10s ()

  Remove 5 secounds from timer.
- void rem1m ()

  Remove 10 secounds from timer.
- void rem5m ()

  Remove 1 minute from timer.
- void rem10m ()

  Remove 5 minutes from timer.

- void resetTimer ()

  Remove 10 minutes from timer.
- void setTimer (bool stat)

  Reset timer.
- void initSerial ()

  Set timer status.
- void sPrint (char ∗abc)

  Inicia a serial.
- void sPrintLn (char ∗abc)

  Print string.
- void sPrintStr (String abc)

  Print string with new line.
- void sPrintNbr (int nbr)

  Print string.
- void sPrintLnStr (String abc)

  Print number.
- void sPrintLnNbr (int nbr)

  Print string with new line.
- void test_IO ()

  Print number with new line.
- bool initWIFI ()

  Initiate WiFi.
- String wifiQuality ()

  Inicia o wifi.

2.3.1 Detailed Description

Header file for the MD project.

Author

Sérgio Carmo

Version

1.0

2.3.2 Macro Definition Documentation

2.3.2.1 ALARM_TIME_OFF

```
#define ALARM_TIME_OFF 200
```

2.3.2.2 ALARM_TIME_ON

```
#define ALARM_TIME_ON 600
```

Processor.

TIMERS Defining timers for the alarm indicator blinking

2.3.2.3 WIFI_MODE_OPTIONS

```
#define WIFI_MODE_OPTIONS 3
```

Testa as entradas e saidas.

WIFI SERVICE WIFI Functions calls

2.3.3 Function Documentation

2.3.3.1 add10m()

```
void add10m ()
```

Add 5 minutes to timer.

Add 5 minutes to timer. Here is the call graph for this function: Here is the caller graph for this function:

2.3.3.2 add10s()

```
void add10s ()
```

Add 5 secounds to timer.

Add 5 secounds to timer. Here is the call graph for this function: Here is the caller graph for this function:

2.3.3.3 add1m()

```
void add1m ()
```

Add 10 secounds to timer.

Add 10 secounds to timer. Here is the call graph for this function: Here is the caller graph for this function:

2.3.3.4 add1s()

```
void add1s ()
```

Get timer status.

Get timer status. Here is the call graph for this function: Here is the caller graph for this function:

2.3.3.5 add5m()

```
void add5m ()
```

Add 1 minute to timer.

Add 1 minute to timer. Here is the call graph for this function: Here is the caller graph for this function:

2.3.3.6 add5s()

```
void add5s ()
```

Add 1 secound to timer.

Add 1 secound to timer. Here is the call graph for this function: Here is the caller graph for this function:

2.3.3.7 getAutoModeWeb()

```
bool getAutoModeWeb ()
```

Set resistor.

Set resistor.

Get the Ind Max object

This function returns the state of the max indicator.

Returns

true

false

Here is the caller graph for this function:

2.3.3.8 getPumpWeb()

```
bool getPumpWeb ()
```

Get auto mode.

Get auto mode.

This function returns the state of the water pump.

Returns

true

false

Here is the caller graph for this function:

2.3.3.9 getResistorWeb()

```
bool getResistorWeb ()
```

Get valve water out.

Get valve water out.

This function returns the state of the min indicator.

Returns

true

false

Here is the caller graph for this function:

2.3.3.10 getTimerHour()

```
int getTimerHour ()
```

Get timer minute.

Get timer minute.

Returns

int

Return the timer hours left in the timerHere is the call graph for this function: Here is the caller graph for this function:

2.3.3.11 getTimerMinute()

```
int getTimerMinute ()
```

Get timer secound.

Get timer secound.

Returns

int

Return the timer minutes left in the timerHere is the call graph for this function: Here is the caller graph for this function:

2.3.3.12 getTimerSecound()

```
int getTimerSecound ()
```

Timer for web operations.

Timer for web operations.

Returns

int

Return the timer seconds left in the timer
Here is the call graph for this function: Here is the caller graph for this function:

2.3.3.13 getTimerStatus()

```
bool getTimerStatus ()
```

Get timer hour.

Get timer hour.

Note

This function is called by the processor function in AP_Destiller.cpp

Returns

true

false

Return the timer status (ON/OFF)Here is the caller graph for this function:

2.3.3.14 getValv_Water_InWeb()

```
bool getValv_Water_InWeb ()
```

Get pump state.

Get pump state.

This function returns the state of the alarm indicator.

Returns

true

false

Here is the caller graph for this function:

2.3.3.15 getValv_Water_OutWeb()

```
bool getValv_Water_OutWeb ()
```

Get valve water in.

Get valve water in.

This function returns the state of the water out valve.

Returns

true

false

Here is the caller graph for this function:

2.3.3.16 handleHour()

```
String handleHour ()
```

Handle timer.

Handle timer.

Returns

String

Note

This function returns the timer hour

Here is the call graph for this function: Here is the caller graph for this function:

2.3.3.17 handleMinute()

```
String handleMinute ()
```

Handle hour.

Handle hour.

Returns

String

Note

This function returns the timer minute

Here is the call graph for this function: Here is the caller graph for this function:

2.3.3.18 handleSecound()

```
String handleSecound ()
```

Handle minute.

Handle minute.

Returns

String

Note

This function returns the timer secound

Here is the call graph for this function: Here is the caller graph for this function:

2.3.3.19 handleState()

```
String handleState (
            const String & state)
```

Handle timer control.

Handle timer control.

Handle State

Parameters

state	

Returns

String

Note

This function handles the state of the variable (auto, manual, pump, waterOut, resistor, waterIn, waterMax, waterMin, waterAlarm)

This function returns the state of the variable (auto, manual, pump, waterOut, resistor, waterIn, waterMax, waterMin, waterAlarm)

Get the state of the variable (auto, manual, pump, waterOut, resistor, waterIn, waterMax, waterMin, waterAlarm)

Get the auto mode indicator state

Get the manual mode indicator state

Get the pump state

Get the water out valve state

Get the resistor state

Get the water in valve state

Get the water max state

Get the water min state

Get the water alarm state

Return nothing if the state is not foundHere is the call graph for this function: Here is the caller graph for this function:

2.3.3.20 handleTimeAdjust()

```
String handleTimeAdjust (
            const String & adjustment)
```

Handle wifi IP.

Handle wifi IP.

Handle Time Adjust

Parameters

adjustment	

Returns

String

Note

This function adjusts the timer

This function returns the adjustment message for the timer (add1s, add5s, add10s, add1m, add5m, add10m, rem1s, rem5s, rem10s, rem1m, rem5m, rem10m)

Adjust the timer based on the adjustment parameter

Add 1 secound to timer

Add 5 secounds to timer

Add 10 secounds to timer

Add 1 minute to timer

Add 5 minutes to timer

Add 10 minutes to timer

Remove 1 secound from timer

Remove 5 secounds from timer

Remove 10 secounds from timer

Remove 1 minute from timer

Remove 5 minutes from timer

Remove 10 minutes from timer

Return the adjustment message for the timer (add1s, add5s, add10s, add1m, add5m, add10m, rem1s, rem5s, rem10s, rem1m, rem5m, rem10m)Here is the call graph for this function: Here is the caller graph for this function:

2.3.3.21 handleTimer()

```
String handleTimer ()
```

Get resistor.

ROUTES AND HANDLERS /// Handlers

Get resistor.

Timer handlers

Handle Timer

Returns

String

Note

This function handles the timer

This function returns the timer in the format HH:MM:SS

Here is the call graph for this function: Here is the caller graph for this function:

2.3.3.22 handleTimerControl()

```
String handleTimerControl (
            const String & control)
```

Handle time adjustment.

Handle time adjustment.

Handle Timer Control

Parameters

control	

Returns

String

Note

This function controls the timer

This function returns the control message for the timer (start, stop, reset)

Control the timer based on the control parameter (start, stop, reset)

Start the timer

Stop the timer

Reset the timer

Return the control message for the timer (start, stop, reset)Here is the call graph for this function: Here is the caller graph for this function:

2.3.3.23 handleToggle()

```
String handleToggle (
            const String & toggle)
```

Handle state.

Handle state.

Handle Toggle

Parameters

toggle	

Returns

String

Note

This function toggles the variable (auto, pump, waterOut, resistor, waterIn)

This function returns the toggle message for the variable (auto, pump, waterOut, resistor, waterIn)

Toggle the variable (auto, pump, waterOut, resistor, waterIn)

Toggle the auto mode indicator state

Toggle the pump state

Toggle the water out valve state

Toggle the water in valve state

Toggle the resistor state

Return the toggle message for the variable (auto, pump, waterOut, resistor, waterIn)Here is the call graph for this function: Here is the caller graph for this function:

2.3.3.24 handleWifiIP()

```
String handleWifiIP ()
```

Handle wifi TX.

Handle wifi TX.

Returns

String

Note

This function returns the WiFi IP address

Here is the caller graph for this function:

2.3.3.25 handleWifiQuality()

```
String handleWifiQuality ()
```

Handle secound.

Handle secound.

Handle Wifi Quality

Returns

String

Note

This function returns the WiFi quality in percentage

Here is the call graph for this function: Here is the caller graph for this function:

2.3.3.26 handleWifiSSID()

```
String handleWifiSSID ()
```

Handle wifi quality.

Handle wifi quality.

Returns

String

Note

This function returns the WiFi SSID

Here is the caller graph for this function:

2.3.3.27 handleWifiTX()

```
String handleWifiTX ()
```

Handle wifi SSID.

Handle wifi SSID.

Returns

String

Note

This function returns the WiFi TX power

Here is the caller graph for this function:

2.3.3.28 indicatorsManagement()

```
void indicatorsManagement ()
```

Toggle auto indicator.

INDICATORS BLOCK

Set the maximum water level indicator

Set the minimum water level indicator

Set the alarm indicator

Set the manual mode indicatorHere is the call graph for this function: Here is the caller graph for this function:

2.3.3.29 initSerial()

```
void initSerial ()
```

Set timer status.

SERIAL Serial Functions calls

Set timer status.

Initialize Serial Communications with the default baud rate of 9600 bps

Note

> This function is used to initialize the Serial Communications with the default baud rate of 9600 bps
>
> This function is called in the setup() function of the main program
>
> This function is used to print the initial message to the Serial Terminal

Initialize Serial Communications with the default baud rate of 9600 bpsHere is the call graph for this function: Here is the caller graph for this function:

2.3.3.30 initWIFI()

```
bool initWIFI ()
```

Initiate WiFi.

Returns

> bool

Note

> This function initiates the WiFi connection

SoftAP flag to create a Soft Access Point only once

WIFI_MODE_OPTIONS

1 - Connect to local WiFi network only

2 - Create a local AP (SoftAP)

3 - Both (Connect to local WiFi network and create a local AP)

Initiate WiFi

Initiate WiFi in AP mode only

Initiate WiFi

Initiate WiFi in AP modeHere is the call graph for this function: Here is the caller graph for this function:

2.3.3.31 modeManagement()

```
void modeManagement ()
```

Indicators management.

MACHINE MODE MANAGEMENT BLOCK // If the machine is in manual mode, turn off the auto mode indicator

Toggle the auto mode

Set the auto mode indicator to the web value of the auto mode

Set the auto mode indicator to OFF

Set the auto mode to OFF in the web interface

If the machine is in manual mode, turn off the timer

Turn off the timerHere is the call graph for this function: Here is the caller graph for this function:

2.3.3.32 processor()

```
String processor (
             const String & var)
```

Handle toggle.

Handle toggle.

Processor

Parameters

var	

Returns

String

Note

This function processes the variables for the web server requests

This function is called by the web server handle function (server.on)

This function returns the variable value for the web server request

Process the variable for the web server request (var) and return the variable value

Timer variable request

Return the timer value in the format HH:MM:SS

WaterMax variable request

Return the waterMax value

WaterMin variable request

Return the waterMin value

WaterAlarm variable request

Return the waterAlarm value

Auto variable request

Return the auto value

Manual variable request

Return the manual value

Pump variable request

Return the pump value

WaterOut variable request

Return the waterOut value

Resistor variable request

Return the resistor value

WaterIn variable request

Return the waterIn value

WifiQuality variable request

Return the wifiQuality value

WifiSSID variable request

Return the wifiSSID value

WifiTX variable request

Return the wifiTX value

WifiIP variable request

Return the wifiIP value

TimerHour variable request

Return the timerHour value

TimerMinute variable request

Return the timerMinute value

TimerSecound variable request

Return the timerSecound value

Return nothing if the variable is not foundHere is the call graph for this function: Here is the caller graph for this function:

2.3.3.33 rem10m()

```
void rem10m ()
```

Remove 5 minutes from timer.

Remove 5 minutes from timer. Here is the call graph for this function: Here is the caller graph for this function:

2.3.3.34 rem10s()

```
void rem10s ()
```

Remove 5 secounds from timer.

Remove 5 secounds from timer. Here is the call graph for this function: Here is the caller graph for this function:

2.3.3.35 rem1m()

```
void rem1m ()
```

Remove 10 secounds from timer.

Remove 10 secounds from timer. Here is the call graph for this function: Here is the caller graph for this function:

2.3.3.36 rem1s()

```
void rem1s ()
```

Add 10 minutes to timer.

Add 10 minutes to timer. Here is the call graph for this function: Here is the caller graph for this function:

2.3.3.37 rem5m()

```
void rem5m ()
```

Remove 1 minute from timer.

Remove 1 minute from timer. Here is the call graph for this function: Here is the caller graph for this function:

2.3.3.38 rem5s()

```
void rem5s ()
```

Remove 1 secound from timer.

Remove 1 secound from timer. Here is the call graph for this function: Here is the caller graph for this function:

2.3.3.39 resetTimer()

```
void resetTimer ()
```

Remove 10 minutes from timer.

Remove 10 minutes from timer. Here is the call graph for this function: Here is the caller graph for this function:

2.3.3.40 setAutoModeWeb()

```
void setAutoModeWeb (
            bool state)
```

Toggle resistor.

Toggle resistor.

Set the Auto Mode object

This function sets the state of the auto mode.

Parameters

state	

Returns

void

Static variable to store the old state

Set the auto indicator to ON

Set the old state to the current state

Set the auto indicator to OFF

Set the old state to the current stateHere is the caller graph for this function:

2.3.3.41 setPumpWeb()

```
void setPumpWeb (
            bool state)
```

Set auto mode.

Set auto mode.

This function sets the state of the pump.

Parameters

state	

Returns

void

Static variable to store the old state

Set the pump to ON

Set the old state to the current state

Set the pump to OFF

Set the old state to the current stateHere is the caller graph for this function:

2.3.3.42 setResistorWeb()

```
void setResistorWeb (
            bool state)
```

Set valve water out.

Set valve water out.

This function sets the state of the resistor.

Parameters

state	

Returns

void

Static variable to store the old state

Set the resistor to ON

Set the old state to the current state

Set the resistor to OFF

Set the old state to the current stateHere is the caller graph for this function:

2.3.3.43 setTimer()

```
void setTimer (
            bool stat)
```

Reset timer.

Reset timer.

Parameters

stat	

Set the timer status (ON/OFF)Here is the call graph for this function: Here is the caller graph for this function:

2.3.3.44 setValveWaterInWeb()

```
void setValveWaterInWeb (
            bool state)
```

Set pump state.

Set pump state.

This function sets the state of the water in valve.

Parameters

state	

Returns

void

Static variable to store the old state

Set the water in valve to ON

Set the old state to the current state

Set the water in valve to OFF

Set the old state to the current stateHere is the caller graph for this function:

2.3.3.45 setValveWaterOutWeb()

```
void setValveWaterOutWeb (
            bool state)
```

Set valve water in.

Set valve water in.

This function sets the state of the water out valve.

Parameters

state	

Returns

void

Static variable to store the old state

Set the water out valve to ON

Set the old state to the current state

Set the water out valve to OFF

Set the old state to the current stateHere is the caller graph for this function:

2.3.3.46 sPrint()

```
void sPrint (
             char * abc)
```

Inicia a serial.

Inicia a serial.

Print a string to the Serial Terminal

Parameters

abc	

Note

This function is used to print a string to the Serial Terminal with no line break

2.3.3.47 sPrintLn()

```
void sPrintLn (
             char * abc)
```

Print string.

Print string.

Parameters

abc	

Note

This function is used to print a string to the Serial Terminal with a line break

2.3.3.48 sPrintLnNbr()

```
void sPrintLnNbr (
             int nbr)
```

Print string with new line.

Print string with new line.

Parameters

nbr	

Note

This function is used to print a number to the Serial Terminal with a line break

Here is the caller graph for this function:

2.3.3.49 sPrintLnStr()

```
void sPrintLnStr (
             String abc)
```

Print number.

Print number.

Parameters

abc	

Note

This function is used to print a string to the Serial Terminal with a line break

Here is the caller graph for this function:

2.3.3.50 sPrintNbr()

```
void sPrintNbr (
             int nbr)
```

Print string.

Print string.

Parameters

nbr	

Note

This function is used to print a number to the Serial Terminal with no line break

Here is the caller graph for this function:

2.3.3.51 sPrintStr()

```
void sPrintStr (
             String abc)
```

Print string with new line.

Print string with new line.

Parameters

abc	

Note

This function is used to print a string to the Serial Terminal with no line break

Here is the caller graph for this function:

2.3.3.52 test_IO()

```
void test_IO ()
```

Print number with new line.

Test I/Os

Print number with new line.

Returns

void

Test I/O Indicador Maximo

Test I/O Indicador Minimo

Test I/O Indicador Alarme

Test I/O Indicador Manual

Test I/O Indicador Auto

Test I/O Indicador Valvula Entrada

Test I/O Indicador Valvula Saida

Test I/O Indicador ResistorHere is the call graph for this function: Here is the caller graph for this function:

2.3.3.53 toggleAutoMode()

```
void toggleAutoMode ()
```

MiddleWare Header.

I/O Map Header Function prototypes

MiddleWare Header.

Note

This function is used to toggle the Auto Mode of the Machine using a switch connected to the ESP32 GPIO 4 (D4)

current state of the button

previous state of the button

current state of the button

the last time the output pin was toggled

the debounce time; increase if the output flickers

read the state of the switch into a local variable

If the switch changed, due to noise or pressing

reset the debouncing timer

if the switch value has been stable for a while

if the button state has changed

save the new state

if the button state is HIGH

Toggle Auto Mode

save the current state as the last state, for next time through the loopHere is the call graph for this function: Here is the caller graph for this function:

2.3.3.54 toggleAutoModeWeb()

```
void toggleAutoModeWeb ()
```

Update outputs for the web interface.

Update outputs for the web interface.

Toggle Auto Mode Web

This function toggles the auto mode.

Returns

void

Get the auto indicator state

Toggle the auto indicator state

Set the auto indicator state

Toggle the auto indicator state

Set the auto indicator stateHere is the call graph for this function: Here is the caller graph for this function:

2.3.3.55 toggleIndAuto()

```
void toggleIndAuto ()
```

Toggle auto mode.

Toggle auto mode.

Note

This function is used to toggle the Auto Mode of the Machine

Here is the call graph for this function: Here is the caller graph for this function:

2.3.3.56 togglePumpWeb()

```
void togglePumpWeb ()
```

Toggle auto mode.

Toggle auto mode.

This function toggles the pump.

Returns

void Toggle Pump

Get the pump state

Toggle the pump state

Set the pump state

Print the pump state

Print the pump state

Toggle the pump state

Set the pump state

Print the pump state

Print the pump stateHere is the call graph for this function: Here is the caller graph for this function:

2.3.3.57 toggleResistorWeb()

```
void toggleResistorWeb ()
```

Toggle valve water out.

Toggle valve water out.

This function toggles the resistor.

Returns

void Toggle Resistor

Get the resistor state

Toggle the resistor state

Set the resistor state

Toggle the resistor state

Set the resistor stateHere is the call graph for this function: Here is the caller graph for this function:

2.3.3.58 toggleValveWaterInWeb()

```
void toggleValveWaterInWeb ()
```

Toggle pump state.

Toggle pump state.

This function toggles the water in valve.

Returns

void Toggle Valve Water In

Get the water in valve state

Toggle the water in valve state

Set the water in valve state

Toggle the water in valve state

Set the water in valve stateHere is the call graph for this function: Here is the caller graph for this function:

2.3.3.59 toggleValveWaterOutWeb()

```
void toggleValveWaterOutWeb ()
```

Toggle valve water in.

Toggle valve water in.

This function toggles the water out valve.

Returns

void Toggle Valve Water Out

Get the water out valve state

If the water out valve state is OFF and the manual mode is ON

Toggle the water out valve state

Set the water out valve state

Toggle the water out valve state

Set the water out valve stateHere is the call graph for this function: Here is the caller graph for this function:

2.3.3.60 updateOutputsWeb()

```
void updateOutputsWeb ()
```

Mode management.

Mode management.

FUNCTIONS BLOCK

Update Outputs Web

This function updates the outputs for the web interface.

Returns

void

Update the resistor state

Update the pump state

Update the water out valve state

Update the water in valve state

Update the alarm state

Update the water min state

Update the water max state

Update the manual mode state

outputVarsImageWeb[7] = getIndAuto(); /// Update the auto indicator stateHere is the call graph for this function: Here is the caller graph for this function:

2.3.3.61 webTimer()

```
long webTimer (
             String op,
             int16_t amount)
```

Timer Functions calls.

Timer Functions calls.

This function is responsible for managing the timer operations.

Parameters

op	
amount	

Returns

unsigned long

Timer operations to add, remove or reset the timer value in miliseconds

Add time to the timer

Add 1 second

Add 5 seconds

Add 10 seconds

Add 1 minute

Add 5 minutes

Add 10 minutes

Remove time from the timer

Remove 1 second

Remove 5 seconds

Remove 10 seconds

Remove 1 minute

Remove 5 minutes

Remove 10 minutes

Reset the timer to 0

Timer management to decrease the timer every second and turn it off when it reaches 0 seconds left

Decrease the timer every second if the timer is ON

Turn off the timer when it reaches 0 seconds left

Only send updates at the defined interval

Send the timer to the web interfaceHere is the call graph for this function: Here is the caller graph for this function:

2.3.3.62 wifiQuality()

```
String wifiQuality ()
```

Inicia o wifi.

Inicia o wifi.

Returns

String

Note

This function returns the WiFi quality in percentage

Get the Received Signal Strength Indicator (RSSI) in dBm

Quality in percentage

Quality in string format

Last quality in percentage

Last RSSI in dBm

If the last RSSI is equal to the current RSSI

Return the quality in string format with RSSI and quality in percentage values

Calculate the quality in percentage

If the RSSI is less than or equal to -100 dBm

Set the quality to 0%

If the RSSI is greater than or equal to -5 dBm

Set the quality to 100%

If the RSSI is between -100 dBm and -5 dBm

Calculate the quality in percentage

Return the quality in string format with RSSI and quality in percentage values

Send the quality in percentage to the web server as a JSON object

Return the quality in string format with RSSI and quality in percentage values

Update the last RSSI with the current RSSIHere is the caller graph for this function:

2.4 MD.h

Go to the documentation of this file.

```


#ifndef _MD_h
#define _MD_h

#include "OS.h"


void toggleAutoMode();
void toggleIndAuto();
void indicatorsManagement();
void modeManagement();

void updateOutputsWeb();
void toggleAutoModeWeb();
void togglePumpWeb();
void toggleValveWaterInWeb();
void toggleValveWaterOutWeb();
void toggleResistorWeb();

void setAutoModeWeb(bool state);
void setPumpWeb(bool state);
void setValveWaterInWeb(bool state);
void setValveWaterOutWeb(bool state);
void setResistorWeb(bool state);

bool getAutoModeWeb();
bool getPumpWeb();
bool getValv_Water_InWeb();
bool getValv_Water_OutWeb();
bool getResistorWeb();

String handleTimer();
String handleHour();
String handleMinute();
String handleSecound();
String handleWifiQuality();
String handleWifiSSID();
String handleWifiTX();
String handleWifiIP();
String handleTimeAdjust(const String &adjustment);
String handleTimerControl(const String &control);
String handleState(const String &state);
String handleToggle(const String &toggle);
String processor(const String &var);


#define ALARM_TIME_ON 600
#define ALARM_TIME_OFF 200

long webTimer(String op, int16_t amount);
int getTimerSecound();
int getTimerMinute();
int getTimerHour();
bool getTimerStatus();
void add1s();
void add5s();
void add10s();
void add1m();
void add5m();
void add10m();
void rem1s();
void rem5s();
void rem10s();
void rem1m();
void rem5m();
void rem10m();
void resetTimer();
void setTimer(bool stat);


void initSerial();
void sPrint(char *abc);
void sPrintLn(char *abc);
void sPrintStr(String abc);
void sPrintNbr(int nbr);
void sPrintLnStr(String abc);
void sPrintLnNbr(int nbr);

void test_IO();


```

```
#define WIFI_MODE_OPTIONS 3
bool initWIFI();
String wifiQuality();

#endif
```

2.5 C:/workspace/Projeto/InteligentDestiller/include/OS.h File Reference

Header file for the OS project.

```
#include "WiFiManager.h"
#include "esp_netif.h"
#include "ESPAsyncWebServer.h"
#include "SPIFFS.h"
#include "MD.h"
```

Include dependency graph for OS.h: This graph shows which files directly or indirectly include this file:

Macros

- #define DEBUGlog 0

  Debug mode.
- #define PIN_SW_AUTO 32

  I/O Map Header.
- #define PIN_SALARM 25
- #define PIN_SW_MAN 26
- #define PIN_SMIN 27
- #define PIN_SMAX 14
- #define PIN_INPUT_COUNT 5
- #define POS_SW_MAN 0
- #define POS_SMIN 1
- #define POS_SMAX 2
- #define POS_SW_AUTO 3
- #define POS_SALARM 4
- #define PIN_RAQ 23

  Output Pins Definitions.
- #define PIN_BMB 22
- #define PIN_VALV_WATER_OUT 21
- #define PIN_VALV_WATER_IN 19
- #define PIN_IND_ALARM 18
- #define PIN_IND_MIN 05
- #define PIN_IND_MAX 04
- #define PIN_IND_AUTO 15
- #define PIN_IND_MAN 33
- #define PIN_OUTPUT_COUNT 9
- #define POS_RAQ 0
- #define POS_BMB 1
- #define POS_VALV_WATER_OUT 2
- #define POS_VALV_WATER_IN 3
- #define POS_IND_ALARM 4
- #define POS_IND_MIN 5
- #define POS_IND_MAX 6
- #define POS_IND_AUTO 7
- #define POS_IND_MAN 8
- #define ON HIGH

  ON/OFF Definitions for I/O pins (HIGH/LOW)
- #define OFF LOW

Functions

- void loop2 (void ∗pvParameters)

  MAIN.
- void initMultiCore ()

  Main loop to run in core 1.
- void initFS ()

  Inicia o segundo core.
- void initPinsOutputs ()

  Inicia o sistema de arquivos.
- void initPinsInputs ()

  Inicia os outputs.
- bool getWaterMax ()

  Inicia os inputs.
- bool getWaterMin ()

  Retorna estado do sensor de agua de maximo.
- bool getAlarm ()

  Retorna estado do sensor de agua de minimo.
- bool getAlarmIND ()

  Retorna estado do alarme.
- bool getIndAuto ()

  Retorna estado do indicador de alarme.
- bool getAutoModeSW ()

  Retorna estado do modo auto.
- bool getManualMode ()

  Retorna estado do interruptor auto.
- bool getResistor ()

  Retorna estado do botao manual.
- bool getValv_Water_In ()

  Retorna estado do resistor.
- bool getValv_Water_Out ()

  Retorna estado da entrada de agua.
- bool getPump ()

  Retorna estado da valvula de vapor.
- void setIndMax (bool state)

  Retorna estado da bomba de agua.
- void setIndMin (bool state)

  Liga/desliga o indicador luminoso nivel agua maximo.
- void setIndAlarm (bool state)

  Liga/desliga o indicador luminoso nivel agua minimo.
- void setIndMan (bool state)

  Liga/desliga o indicador luminoso alarme.
- void setPump (bool state)

  Liga/desliga o indicador luminoso modo manual.
- void setValveWaterIn (bool state)

  Liga/desliga o indicador luminoso da bomba de agua.
- void setValveWaterOut (bool state)

  Liga/desliga o indicador luminoso da valvula 1.
- void setResistor (bool state)

  Liga/desliga o indicador luminoso da valvula 2.
- void setIndAuto (bool state1)

Liga/desliga o indicador luminoso da resistencia.

- void writeOutputs ()

  Liga/desliga o modo auto.
- void readInputs ()

  Atualiza as saidas.

Variables

- AsyncWebSocket ws

  Libraries.
- AsyncWebServer server

  WebSocket Server.
- bool DEBUG

  WebServer.

2.5.1 Detailed Description

Header file for the OS project.

Author

Sérgio Carmo

Version

1.0

2.5.2 Macro Definition Documentation

2.5.2.1 DEBUGlog

```
#define DEBUGlog 0
```

Debug mode.

2.5.2.2 OFF

```
#define OFF LOW
```

2.5.2.3 ON

```
#define ON HIGH
```

ON/OFF Definitions for I/O pins (HIGH/LOW)

2.5.2.4 PIN_BMB

```
#define PIN_BMB 22
```

2.5.2.5 PIN_IND_ALARM

```
#define PIN_IND_ALARM 18
```

2.5.2.6 PIN_IND_AUTO

```
#define PIN_IND_AUTO 15
```

2.5.2.7 PIN_IND_MAN

```
#define PIN_IND_MAN 33
```

2.5.2.8 PIN_IND_MAX

```
#define PIN_IND_MAX 04
```

2.5.2.9 PIN_IND_MIN

```
#define PIN_IND_MIN 05
```

2.5.2.10 PIN_INPUT_COUNT

```
#define PIN_INPUT_COUNT 5
```

2.5.2.11 PIN_OUTPUT_COUNT

```
#define PIN_OUTPUT_COUNT 9
```

2.5.2.12 PIN_RAQ

```
#define PIN_RAQ 23
```

Output Pins Definitions.

2.5.2.13 PIN_SALARM

```
#define PIN_SALARM 25
```

2.5.2.14 PIN_SMAX

```
#define PIN_SMAX 14
```

2.5.2.15 PIN_SMIN

```
#define PIN_SMIN 27
```

2.5.2.16 PIN_SW_AUTO

```
#define PIN_SW_AUTO 32
```

I/O Map Header.

Input Pins Definitions

2.5.2.17 PIN_SW_MAN

```
#define PIN_SW_MAN 26
```

2.5.2.18 PIN_VALV_WATER_IN

```
#define PIN_VALV_WATER_IN 19
```

2.5.2.19 PIN_VALV_WATER_OUT

```
#define PIN_VALV_WATER_OUT 21
```

2.5.2.20 POS_BMB

```
#define POS_BMB 1
```

2.5.2.21 POS_IND_ALARM

```
#define POS_IND_ALARM 4
```

2.5.2.22 POS_IND_AUTO

```
#define POS_IND_AUTO 7
```

2.5.2.23 POS_IND_MAN

```
#define POS_IND_MAN 8
```

2.5.2.24 POS_IND_MAX

```
#define POS_IND_MAX 6
```

2.5.2.25 POS_IND_MIN

```
#define POS_IND_MIN 5
```

2.5.2.26 POS_RAQ

```
#define POS_RAQ 0
```

2.5.2.27 POS_SALARM

```
#define POS_SALARM 4
```

2.5.2.28 POS_SMAX

```
#define POS_SMAX 2
```

2.5.2.29 POS_SMIN

```
#define POS_SMIN 1
```

2.5.2.30 POS_SW_AUTO

```
#define POS_SW_AUTO 3
```

2.5.2.31 POS_SW_MAN

```
#define POS_SW_MAN 0
```

2.5.2.32 POS_VALV_WATER_IN

```
#define POS_VALV_WATER_IN 3
```

2.5.2.33 POS_VALV_WATER_OUT

```
#define POS_VALV_WATER_OUT 2
```

2.5.3 Function Documentation

2.5.3.1 getAlarm()

```
bool getAlarm ()
```

Retorna estado do sensor de agua de minimo.

Retorna estado do sensor de agua de minimo.

This function returns the state of the alarm.

Returns

true

false

Here is the caller graph for this function:

2.5.3.2 getAlarmIND()

```
bool getAlarmIND ()
```

Retorna estado do alarme.

Retorna estado do alarme.

This function returns the state of the alarm.

Returns

true

false

Here is the caller graph for this function:

2.5.3.3 getAutoModeSW()

```
bool getAutoModeSW ()
```

Retorna estado do modo auto.

Retorna estado do modo auto.

This function returns the state of the auto mode switch.

Returns

true

false

Here is the caller graph for this function:

2.5.3.4 getIndAuto()

```
bool getIndAuto ()
```

Retorna estado do indicador de alarme.

Retorna estado do indicador de alarme.

This function returns the state of the max indicator.

Returns

true

false

Here is the caller graph for this function:

2.5.3.5 getManualMode()

```
bool getManualMode ()
```

Retorna estado do interruptor auto.

Retorna estado do interruptor auto.

This function returns the state of the manual mode switch.

Returns

true

false

Here is the caller graph for this function:

2.5.3.6 getPump()

```
bool getPump ()
```

Retorna estado da valvula de vapor.

Retorna estado da valvula de vapor.

This function returns the state of the water pump.

Returns

true

false

Here is the caller graph for this function:

2.5.3.7 getResistor()

```
bool getResistor ()
```

Retorna estado do botao manual.

Retorna estado do botao manual.

This function returns the state of the min indicator.

Returns

true

false

Here is the caller graph for this function:

2.5.3.8 getValv_Water_In()

```
bool getValv_Water_In ()
```

Retorna estado do resistor.

Retorna estado do resistor.

This function returns the state of the alarm indicator.

Returns

true

false

Here is the caller graph for this function:

2.5.3.9 getValv_Water_Out()

```
bool getValv_Water_Out ()
```

Retorna estado da entrada de agua.

Retorna estado da entrada de agua.

This function returns the state of the water out valve.

Returns

true

false

Here is the caller graph for this function:

2.5.3.10 getWaterMax()

```
bool getWaterMax ()
```

Inicia os inputs.

Inicia os inputs.

Get the Water Max object

This function returns the state of the water max sensor.

Returns

true

false

Here is the caller graph for this function:

2.5.3.11 getWaterMin()

```
bool getWaterMin ()
```

Retorna estado do sensor de agua de maximo.

Retorna estado do sensor de agua de maximo.

This function returns the state of the water min sensor.

Returns

true

false

Here is the caller graph for this function:

2.5.3.12 initFS()

```
void initFS ()
```

Inicia o segundo core.

LittleFS

Inicia o segundo core.

Initialize File System (SPIFFS)

This function is responsible for initializing the File System (SPIFFS).

Note

Initialize SPIFFS

Returns

void

```
initFS();
```

Initialize SPIFFSHere is the call graph for this function: Here is the caller graph for this function:

2.5.3.13 initMultiCore()

```
void initMultiCore ()
```

Main loop to run in core 1.

Multicore

Main loop to run in core 1.

<Function to initiate the 2nd core> Implements the 2nd core option to run 'loop2' function

Function to implement the task

Name of the task

Stack size in words

Task input parameter

Priority of the task

Task handle.

Core where the task should runHere is the call graph for this function: Here is the caller graph for this function:

2.5.3.14 initPinsInputs()

```
void initPinsInputs ()
```

Inicia os outputs.

Inicia os outputs.

Configure GPIOs as INPUTs

This function configures the GPIOs as INPUTs. The GPIOs are used to read the state of the switches and sensors. on/off sw

min sensor

max sensor

push sw

alarmHere is the caller graph for this function:

2.5.3.15 initPinsOutputs()

```
void initPinsOutputs ()
```

Inicia o sistema de arquivos.

I/O I/O Functions calls

Inicia o sistema de arquivos.

Configure GPIOs as OUTPUTs

This function configures the GPIOs as OUTPUTs. The GPIOs are used to control the indicators, valves, pump, and resistor. resistor

water in valve

water out valve

water pump

min indicator

max indicator

alarm indicator

manual indicator

auto indicatorHere is the caller graph for this function:

2.5.3.16 loop2()

```
void loop2 (
            void * pvParameters)
```

MAIN.

MAIN.

This function is responsible for running the second core operation.

Note

Initiate WIFI
Cleanup clients
WebTimer function

See also

initWIFI()
ws.cleanupClients()
webTimer()

Parameters

pvParameters	

Main loop

Cleanup clients

WebTimer functionHere is the call graph for this function: Here is the caller graph for this function:

2.5.3.17 readInputs()

```
void readInputs ()
```

Atualiza as saidas.

Atualiza as saidas.

Read ESP32 GPIOs and stores in the image array

Read all inputs and store in the image array

This function reads all the inputs and stores them in the image array. The image array is used to store the current state of the inputs and outputs. PIN_INPUT_COUNT = 5

Read the input from the GPIOHere is the caller graph for this function:

2.5.3.18 setIndAlarm()

```
void setIndAlarm (
            bool state)
```

Liga/desliga o indicador luminoso nivel agua minimo.

Liga/desliga o indicador luminoso nivel agua minimo.

This function sets the state of the alarm indicator.

Parameters

state	

Returns

void

Old state of the indicator

Condition to debugHere is the call graph for this function: Here is the caller graph for this function:

2.5.3.19 setIndAuto()

```
void setIndAuto (
            bool state)
```

Liga/desliga o indicador luminoso da resistencia.

Liga/desliga o indicador luminoso da resistencia.

This function sets the state of the auto mode.

Parameters

state	

Returns

void

Here is the call graph for this function: Here is the caller graph for this function:

2.5.3.20 setIndMan()

```
void setIndMan (
            bool state)
```

Liga/desliga o indicador luminoso alarme.

Liga/desliga o indicador luminoso alarme.

This function sets the state of the manual indicator.

Parameters

state	

Returns

void

Here is the call graph for this function: Here is the caller graph for this function:

2.5.3.21 setIndMax()

```
void setIndMax (
            bool state)
```

Retorna estado da bomba de agua.

Retorna estado da bomba de agua.

Set the Ind Max object

This function sets the state of the max indicator.

Parameters

state	

Returns

void

Old state of the indicator

Set the max indicator to ON

Condition to debug

Send the max indicator value to the web server

Set the max indicator to OFF

Send the max indicator value to the web serverHere is the call graph for this function: Here is the caller graph for this function:

2.5.3.22 setIndMin()

```
void setIndMin (
            bool state)
```

Liga/desliga o indicador luminoso nivel agua maximo.

Liga/desliga o indicador luminoso nivel agua maximo.

This function sets the state of the min indicator.

Parameters

state	

Returns

void

Old state of the indicator

Set the min indicator to ON

Send the min indicator value to the web server

Set the min indicator to OFF

Send the min indicator value to the web serverHere is the call graph for this function: Here is the caller graph for this function:

2.5.3.23 setPump()

```
void setPump (
            bool state)
```

Liga/desliga o indicador luminoso modo manual.

Liga/desliga o indicador luminoso modo manual.

This function sets the state of the pump.

Parameters

state	

Returns

void

Here is the call graph for this function: Here is the caller graph for this function:

2.5.3.24 setResistor()

```
void setResistor (
            bool state)
```

Liga/desliga o indicador luminoso da valvula 2.

Liga/desliga o indicador luminoso da valvula 2.

This function sets the state of the resistor.

Parameters

state	

Returns

void

Here is the call graph for this function: Here is the caller graph for this function:

2.5.3.25 setValveWaterIn()

```
void setValveWaterIn (
            bool state)
```

Liga/desliga o indicador luminoso da bomba de agua.

Liga/desliga o indicador luminoso da bomba de agua.

This function sets the state of the water in valve.

Parameters

state	

Returns

void

Here is the call graph for this function: Here is the caller graph for this function:

2.5.3.26 setValveWaterOut()

```
void setValveWaterOut (
            bool state)
```

Liga/desliga o indicador luminoso da valvula 1.

Liga/desliga o indicador luminoso da valvula 1.

This function sets the state of the water out valve.

Parameters

state	

Returns

void

Here is the call graph for this function: Here is the caller graph for this function:

2.5.3.27 writeOutputs()

```
void writeOutputs ()
```

Liga/desliga o modo auto.

Liga/desliga o modo auto.

Write all outputs from the image array to the GPIOs

This function writes all the outputs from the image array to the GPIOs. The image array is used to store the current state of the inputs and outputs. PIN_OUTPUT_COUNT = 9

Write the output to the GPIOHere is the caller graph for this function:

2.5.4 Variable Documentation

2.5.4.1 DEBUG

```
bool DEBUG [extern]
```

WebServer.

2.5.4.2 server

```
AsyncWebServer server [extern]
```

WebSocket Server.

2.5.4.3 ws

```
AsyncWebSocket ws [extern]
```

Libraries.

Wifi Libraries WiFi Manager
ESP32 Network Interface Async Libraries Async Web Server SPIFFS Libraries SPI Flash File System MD Header

2.6 OS.h

Go to the documentation of this file.

```

#ifndef _OS_h
#define _OS_h


#include "WiFiManager.h"
#include "esp_netif.h"

#include "ESPAsyncWebServer.h"

#include "SPIFFS.h"

#include "MD.h"

extern AsyncWebSocket ws;
extern AsyncWebServer server;

extern bool DEBUG;
#define DEBUGlog 0


#define PIN_SW_AUTO 32
#define PIN_SALARM 25
#define PIN_SW_MAN 26
#define PIN_SMIN 27
#define PIN_SMAX 14

#define PIN_INPUT_COUNT 5
#define POS_SW_MAN 0
#define POS_SMIN 1
#define POS_SMAX 2
#define POS_SW_AUTO 3
#define POS_SALARM 4

#define PIN_RAQ 23
#define PIN_BMB 22
#define PIN_VALV_WATER_OUT 21
#define PIN_VALV_WATER_IN 19
#define PIN_IND_ALARM 18
#define PIN_IND_MIN 05
#define PIN_IND_MAX 04
#define PIN_IND_AUTO 15
#define PIN_IND_MAN 33

#define PIN_OUTPUT_COUNT 9
#define POS_RAQ 0
#define POS_BMB 1
#define POS_VALV_WATER_OUT 2
#define POS_VALV_WATER_IN 3
#define POS_IND_ALARM 4
#define POS_IND_MIN 5
#define POS_IND_MAX 6
#define POS_IND_AUTO 7
#define POS_IND_MAN 8

#define ON HIGH
#define OFF LOW

void loop2(void *pvParameters);

void initMultiCore();

void initFS();


void initPinsOutputs();
void initPinsInputs();
bool getWaterMax();
bool getWaterMin();
bool getAlarm();
bool getAlarmIND();
bool getIndAuto();
bool getAutoModeSW();
bool getManualMode();
bool getResistor();
bool getValv_Water_In();
bool getValv_Water_Out();
bool getPump();

void setIndMax(bool state);
void setIndMin(bool state);
void setIndAlarm(bool state);
void setIndMan(bool state);
void setPump(bool state);
void setValveWaterIn(bool state);
void setValveWaterOut(bool state);
void setResistor(bool state);
void setIndAuto(bool state1);
void writeOutputs();
void readInputs();

#endif
```

2.7 C:/workspace/Projeto/InteligentDestiller/src/AP_Destiler.cpp File Reference

Application file for the MD project.

`#include "AP.h"`

Include dependency graph for AP_Destiler.cpp:

Functions

- void destiler ()

  Destiler function.

2.7.1 Detailed Description

Application file for the MD project.

Author

Sérgio Carmo

Version

1.0

2.7.2 Function Documentation

2.7.2.1 destiler()

```
void destiler ()
```

Destiler function.

APP Header.

This function is responsible for managing the operation of the machine. BMB_FLAG, RAQ_FLAG BMB_FLAG and RAQ_FLAG are flags that are used to control the operation of the machine. BMB_FLAG is used to control the operation of the water pump. RAQ_FLAG is used to control the operation of the water heater resistor.

Note

BMB_FLAG and RAQ_FLAG are used to control the operation of the machine.

BMB_FLAG is used to control the operation of the water pump.

RAQ_FLAG is used to control the operation of the water heater resistor.

< Debug information logging

Call the indicatorsManagement function to manage the indicators of the machine

Call the modeManagement function to manage the different modes of the machine (Auto and Manual)

WORKING BLOCK

Flag to control the operation of the water pump

If the water level is at the maximum, the water pump is turned off. If the water level is at the minimum or there is an alarm, the water pump is turned on.

Note

If the water level is at the maximum, the water pump is turned off.

If the water level is at the minimum or there is an alarm, the water pump is turned on.

If the water level is at the maximum turn off the water pump

Turn off the water pump

Set the water pump to OFF in the web interface

If the water level is at the minimum or at the alarm level, turn on the water pump

Turn on the water pump

Control of the operation of the water heater resistor

If the water level is at the maximum, the water heater resistor is turned off. If the water level is at the minimum or there is an alarm, the water heater resistor is turned on.

Note

If the water level is at the maximum, the water heater resistor is turned off.

If the water level is at the minimum or there is an alarm, the water heater resistor is turned on. *

If the water level is at the maximum or min or is not at alarm lvl, turn ON the water heater resistor

If the water level is at the alarmlvl, turn OFF the water heater resistor

BMB, RAQ, V_IN, V_OUT BMB, RAQ, V_IN, V_OUT are variables that are used to control the operation of the machine. BMB is used to control the operation of the water pump. RAQ is used to control the operation of the water heater resistor. V_IN is used to control the operation of the water inlet valve. V_OUT is used to control the operation of the water outlet valve.

Note

BMB is used to control the operation of the water pump.

RAQ is used to control the operation of the water heater resistor.

V_IN is used to control the operation of the water inlet valve.

V_OUT is used to control the operation of the water outlet valve.

If the water pump is on and the machine is in auto mode or the water pump is on in manual mode and the water level is not at the maximum, turn on the water pump

If the water heater resistor is on and the machine is in auto mode or the water heater resistor is on in manual mode and the alarm is not on, turn on the water heater resistor

If the water inlet valve is on and the machine is in auto mode or the water inlet valve is on in manual mode and the alarm is not on, turn on the water inlet valve

If the water outlet valve is on and the machine is in auto mode or the water outlet valve is on in manual mode and the alarm is not on, turn on the water outlet valve

Set the Pump object to the BMB variable value Set the Pump object to the BMB variable value.

Note

Set the Pump object to the BMB variable value.

Set the Resistor object to the RAQ variable value Set the Resistor object to the RAQ variable value.

Note

Set the Resistor object to the RAQ variable value.

Set the Valve Water In object Set the Valve Water In object to the V_IN variable value.

Note

Set the Valve Water In object to the V_IN variable value.

Set the Valve Water Out object Set the Valve Water Out object to the V_OUT variable value.

Note

Set the Valve Water Out object to the V_OUT variable value.

Here is the call graph for this function: Here is the caller graph for this function:

2.8 C:/workspace/Projeto/InteligentDestiller/src/AP_main.cpp File Reference

Main file for the MD project.

```
#include "AP.h"
```

Include dependency graph for AP_main.cpp:

Functions

- AsyncWebSocket ws ("/ws")

  Debug mode.
- AsyncWebServer server (80)

  AsyncWebSocket object to handle websocket connections.
- void setup ()

  Create a webserver object that listens for HTTP request on port 80.
- void loop (void)

  Loop function to run the machine operation.
- void loop2 (void *pvParameters)

  Main function to run the second core operation.

Variables

- bool DEBUG = 1

  WebServer.

2.8.1 Detailed Description

Main file for the MD project.

Author

Sérgio Carmo

Version

1.0

2.8.2 Function Documentation

2.8.2.1 loop()

```
void loop (
            void )
```

Loop function to run the machine operation.

This function is responsible for running the machine operation.

Note

Read inputs

Destiler function to operate the machine

Update outputs

See also

readInputs()

destiler()

updateOutputsWeb()

Destiler function to operate the machine

Update outputsHere is the call graph for this function:

2.8.2.2 loop2()

```
void loop2 (
             void * pvParameters)
```

Main function to run the second core operation.

MAIN.

This function is responsible for running the second core operation.

Note

Initiate WIFI

Cleanup clients

WebTimer function

See also

initWIFI()

ws.cleanupClients()

webTimer()

Parameters

pvParameters	

Main loop

Cleanup clients

WebTimer functionHere is the call graph for this function: Here is the caller graph for this function:

2.8.2.3 server()

```
AsyncWebServer server (
             80 )
```

AsyncWebSocket object to handle websocket connections.

2.8.2.4 setup()

```
void setup ()
```

Create a webserver object that listens for HTTP request on port 80.

Setup function to initiate all the necessary functions to run the machine

This function is responsible for initiating all the necessary functions to run the machine.

Note

Initiate all pins as inputs
Initiate all pins as outputs
Initiate Serial communication
Initiate MultiCore
Initiate SPIFFS
Read inputs
Update outputs
Setup Routes
Add handler to the server

Initiate all pins as inputs

Initiate all pins as outputs

Initiate Serial communication (9600 bps)

Test IO

Initiate MultiCore (Second Core)

Initiate SPIFFS (SPI Flash File System)

Read inputs for the first time

Update outputs for the web interface

Initiate WIFI

Setup Routes

Add handler to the server (Websocket)Here is the call graph for this function:

2.8.2.5 ws()

```
AsyncWebSocket ws (
            "/ws" )
```

Debug mode.

AsyncWebSocket object to handle websocket connections

2.8.3 Variable Documentation

2.8.3.1 DEBUG

```
bool DEBUG = 1
```

WebServer.

2.9 C:/workspace/Projeto/InteligentDestiller/src/AP_Web.cpp File Reference

Web Server functions.

```
#include "AP.h"
```

Include dependency graph for AP_Web.cpp:

Functions

- void handleWebSocketMessage (void ∗arg, uint8_t ∗data, size_t len)

  Definitions.
- void onEvent (AsyncWebSocket ∗server, AsyncWebSocketClient ∗client, AwsEventType type, void ∗arg, uint8_t ∗data, size_t len)

  Web Server and WebSocket Server variables and constants initialization function.
- void setupRoutes ()

  Setup Calls.

2.9.1 Detailed Description

Web Server functions.

Author

Sérgio Carmo

Version

1.0

2.9.2 Function Documentation

2.9.2.1 handleWebSocketMessage()

```
void handleWebSocketMessage (
            void * arg,
            uint8_t * data,
            size_t len)
```

Definitions.

Variables and Constants for Web Server and WebSocket Server

Web Server and WebSocket Server variables

These variables are used to create the Web Server and WebSocket Server.

Note

These variables are used to create the Web Server and WebSocket Server.

These variables are used to handle the HTTP GET requests for various resources.

Parameters

server	Web Server
ws	WebSocket Server
processor	Web Server Processor
wsFrameInfo	WebSocket Frame Information
wsMask	WebSocket Mask

Returns

void

WebSocket Frame Information Structure Pointer (info)

Check if the WebSocket message is a text message and the final message in the frame sequence and the length of the message is equal to the length of the frame and the index of the frame is 0

Null-terminate the data to make it a string (char array)

Convert the data to a String object

Print the received WebSocket messageHere is the caller graph for this function:

2.9.2.2 onEvent()

```
void onEvent (
            AsyncWebSocket * server,
            AsyncWebSocketClient * client,
            AwsEventType type,
            void * arg,
            uint8_t * data,
            size_t len)
```

Web Server and WebSocket Server variables and constants initialization function.

This function initializes the Web Server and WebSocket Server variables and constants.

Note

This function initializes the Web Server and WebSocket Server variables and constants.

Parameters

server	
client	
type	
arg	
data	
len	

Handle WebSocket events based on the event type (type)

WebSocket client connected event

Print the IP address of the connected WebSocket client

WebSocket client disconnected event

Print the IP address of the disconnected WebSocket client

WebSocket data event

Handle the WebSocket message data event

WebSocket Pong event (response to a Ping)

WebSocket error eventHere is the call graph for this function: Here is the caller graph for this function:

2.9.2.3 setupRoutes()

```
void setupRoutes ()
```

Setup Calls.

Setup the routes for the web server.

This function defines the routes for handling HTTP GET requests for various resources such as HTML pages, CSS files, JS files, images, time adjustments, timer control, toggle actions, state actions, WiFi handlers, and timer handlers. Set the WebSocket event handler function for the WebSocket Server (ws)

Add the WebSocket handler to the server (server) // Route for root / and index_webButtons.html

Send the index.html file to the client when the root / is requested

Send the index.html file to the client when the root / is requested

Route for CSS files

Send the style.css file to the client when requested

Send the style.css file to the client when requested

Send the style.css file to the client when requested

Route JS files

Send the script_actions.js file to the client when requested

Send the script_get_Data.js file to the client when requested

Send the script_timers.js file to the client when requested

Send the script_timers_Img.js file to the client when requested

Send the script_timers_2.js file to the client when requested

Route for image files

Send the icon.png file to the client when requested

Send the estg_logo.png file to the client when requested

Send the bg.png file to the client when requested

Send the destiller.png file to the client when requested

Send the BMB_ON.png file to the client when requested

Send the BMB_OFF.png file to the client when requested

Send the RAQ_ON.png file to the client when requested

Send the RAQ_OFF.png file to the client when requested

Send the Valve_ON.png file to the client when requested

Send the Valve_OFF.png file to the client when requested

Send the Auto_ON.png file to the client when requested

Send the Auto_OFF.png file to the client when requested

Send the Man_ON.png file to the client when requested

Send the Man_OFF.png file to the client when requested

Send the sMin.png file to the client when requested

Send the sMax.png file to the client when requested

Send the sAlarm.png file to the client when requested

Send the button_ON.png file to the client when requested

Send the button_OFF.png file to the client when requested

Routes for handling time adjustments

Send the response to the client when the /add1s route is requested

Send the response to the client when the /add5s route is requested

Send the response to the client when the /add10s route is requested

Send the response to the client when the /add1m route is requested

Send the response to the client when the /add5m route is requested

Send the response to the client when the /add10m route is requested

Send the response to the client when the /rem1s route is requested

Send the response to the client when the /rem5s route is requested

Send the response to the client when the /rem10s route is requested

Send the response to the client when the /rem1m route is requested

Send the response to the client when the /rem5m route is requested

Send the response to the client when the /rem10m route is requested

Routes for handling timer control

Send the response to the client when the /startTimer route is requested

Send the response to the client when the /stopTimer route is requested

Send the response to the client when the /resetTimer route is requested

Routes for handling toggle actions

Send the response to the client when the /toggleAutoMode route is requested

Send the response to the client when the /togglePump route is requested

Send the response to the client when the /toggleWaterOut route is requested

Send the response to the client when the /toggleWaterIn route is requested

Send the response to the client when the /toggleResistor route is requested

Routes for handling read actions

Send the response to the client when the /readAutoMode route is requested

Send the response to the client when the /readManualMode route is requested

Send the response to the client when the /readPump route is requested

Send the response to the client when the /readWaterOut route is requested

Send the response to the client when the /readResistor route is requested

Send the response to the client when the /readWaterIn route is requested

Send the response to the client when the /readWaterMax route is requested

Send the response to the client when the /readWaterMin route is requested

Send the response to the client when the /readWaterAlarm route is requested

Route for WiFi Handlers

Send the response to the client when the /readWifiQuality route is requested

Send the response to the client when the /readWifiSSID route is requested

Send the response to the client when the /readWifiTX route is requested

Send the response to the client when the /readWifiIP route is requested

Route for Timer Handlers

Send the response to the client when the /readTimer route is requested

Send the response to the client when the /readHour route is requested

Send the response to the client when the /readMinute route is requested

Send the response to the client when the /readSecound route is requested

Start Server

Start the serverHere is the call graph for this function: Here is the caller graph for this function:

2.10 C:/workspace/Projeto/InteligentDestiller/src/MD_IO.cpp File Reference

IO functions for the MD project <This file contains the IO functions for the MD project that are used to control the Machine and its peripherals using the ESP32 GPIOs and the Web Interface of the project.>

```
#include "MD.h"
```

Include dependency graph for MD_IO.cpp:

Functions

- void toggleIndAuto ()

  Toggle the Auto Mode of the Machine.
- void toggleAutoMode ()

  Toggle the Auto Mode of the Machine using a switch connected to the ESP32 GPIO 4 (D4)
- void indicatorsManagement ()

  Toggle auto indicator.
- void modeManagement ()

  Indicators management.

2.10.1 Detailed Description

IO functions for the MD project <This file contains the IO functions for the MD project that are used to control the Machine and its peripherals using the ESP32 GPIOs and the Web Interface of the project.>

Author

Sérgio Carmo

Version

1.0

2.10.2 Function Documentation

2.10.2.1 indicatorsManagement()

```
void indicatorsManagement ()
```

Toggle auto indicator.

INDICATORS BLOCK

Set the maximum water level indicator

Set the minimum water level indicator

Set the alarm indicator

Set the manual mode indicatorHere is the call graph for this function: Here is the caller graph for this function:

2.10.2.2 modeManagement()

```
void modeManagement ()
```

Indicators management.

MACHINE MODE MANAGEMENT BLOCK // If the machine is in manual mode, turn off the auto mode indicator

Toggle the auto mode

Set the auto mode indicator to the web value of the auto mode

Set the auto mode indicator to OFF

Set the auto mode to OFF in the web interface

If the machine is in manual mode, turn off the timer

Turn off the timerHere is the call graph for this function: Here is the caller graph for this function:

2.10.2.3 toggleAutoMode()

```
void toggleAutoMode ()
```

Toggle the Auto Mode of the Machine using a switch connected to the ESP32 GPIO 4 (D4)

MiddleWare Header.

Note

> This function is used to toggle the Auto Mode of the Machine using a switch connected to the ESP32 GPIO 4 (D4)

current state of the button

previous state of the button

current state of the button

the last time the output pin was toggled

the debounce time; increase if the output flickers

read the state of the switch into a local variable

If the switch changed, due to noise or pressing

reset the debouncing timer

if the switch value has been stable for a while

if the button state has changed

save the new state

if the button state is HIGH

Toggle Auto Mode

save the current state as the last state, for next time through the loopHere is the call graph for this function: Here is the caller graph for this function:

2.10.2.4 toggleIndAuto()

```
void toggleIndAuto ()
```

Toggle the Auto Mode of the Machine.

Toggle auto mode.

Note

> This function is used to toggle the Auto Mode of the Machine

Here is the call graph for this function: Here is the caller graph for this function:

2.11 C:/workspace/Projeto/InteligentDestiller/src/MD_Serial.cpp File Reference

Serial functions for the MD project.

```
#include "MD.h"
```

Include dependency graph for MD_Serial.cpp:

Functions

- void initSerial ()

  Initialize Serial Terminal.
- void sPrint (char ∗abc)

  Serial Print Functions.
- void sPrintLn (char ∗abc)

  Print a string to the Serial Terminal with a line break.
- void sPrintStr (String abc)

  Print a string to the Serial Terminal.
- void sPrintLnStr (String abc)

  Print a string to the Serial Terminal with a line break.
- void sPrintNbr (int nbr)

  Print a number to the Serial Terminal.
- void sPrintLnNbr (int nbr)

  Print a number to the Serial Terminal with a line break.

2.11.1 Detailed Description

Serial functions for the MD project.

Author

Sérgio Carmo

Version

1.0

2.11.2 Function Documentation

2.11.2.1 initSerial()

```
void initSerial ()
```

Initialize Serial Terminal.

Set timer status.

Initialize Serial Communications with the default baud rate of 9600 bps

Note

This function is used to initialize the Serial Communications with the default baud rate of 9600 bps

This function is called in the setup() function of the main program

This function is used to print the initial message to the Serial Terminal

Initialize Serial Communications with the default baud rate of 9600 bpsHere is the call graph for this function: Here is the caller graph for this function:

2.11.2.2 sPrint()

```
void sPrint (
             char * abc)
```

Serial Print Functions.

Inicia a serial.

Print a string to the Serial Terminal

Parameters

abc	

Note

> This function is used to print a string to the Serial Terminal with no line break

2.11.2.3 sPrintLn()

```
void sPrintLn (
             char * abc)
```

Print a string to the Serial Terminal with a line break.

Print string.

Parameters

abc	

Note

> This function is used to print a string to the Serial Terminal with a line break

2.11.2.4 sPrintLnNbr()

```
void sPrintLnNbr (
             int nbr)
```

Print a number to the Serial Terminal with a line break.

Print string with new line.

Parameters

nbr	

Note

> This function is used to print a number to the Serial Terminal with a line break

Here is the caller graph for this function:

2.11.2.5 sPrintLnStr()

```
void sPrintLnStr (
             String abc)
```

Print a string to the Serial Terminal with a line break.

Print number.

Parameters

abc	

Note

This function is used to print a string to the Serial Terminal with a line break

Here is the caller graph for this function:

2.11.2.6 sPrintNbr()

```
void sPrintNbr (
            int nbr)
```

Print a number to the Serial Terminal.

Print string.

Parameters

nbr	

Note

This function is used to print a number to the Serial Terminal with no line break

Here is the caller graph for this function:

2.11.2.7 sPrintStr()

```
void sPrintStr (
            String abc)
```

Print a string to the Serial Terminal.

Print string with new line.

Parameters

abc	

Note

This function is used to print a string to the Serial Terminal with no line break

Here is the caller graph for this function:

2.12 C:/workspace/Projeto/InteligentDestiller/src/MD_Test_IOs.cpp File Reference

Test I/Os for the MD project.

```
#include "MD.h"
```

Include dependency graph for MD_Test_IOs.cpp:

Macros

- #define delayT 500

Functions

- void test_IO ()

  Test I/Os.

2.12.1 Detailed Description

Test I/Os for the MD project.

Author

Sérgio Carmo

Version

1.0

2.12.2 Macro Definition Documentation

2.12.2.1 delayT

```
#define delayT 500
```

2.12.3 Function Documentation

2.12.3.1 test_IO()

```
void test_IO ()
```

Test I/Os.

Print number with new line.

Returns

void

Test I/O Indicador Maximo

Test I/O Indicador Minimo

Test I/O Indicador Alarme

Test I/O Indicador Manual

Test I/O Indicador Auto

Test I/O Indicador Valvula Entrada

Test I/O Indicador Valvula Saida

Test I/O Indicador ResistorHere is the call graph for this function: Here is the caller graph for this function:

2.13 C:/workspace/Projeto/InteligentDestiller/src/MD_Timer.cpp File Reference

```
#include "MD.h"
```

Include dependency graph for MD_Timer.cpp:

Functions

- void sendTimer ()

  Send timer to web.
- long webTimer (String op, int16_t amount)

  Web timer function to manage the timer operations in the web interface.
- bool getTimerStatus ()

  Get the Timer Status object to get the timer status (ON/OFF)
- int getTimerSecound ()

  Get the Timer Secound object to get the timer secound.
- int getTimerMinute ()

  Get the Timer Minute object to get the timer minute.
- int getTimerHour ()

  Get the Timer Hour object to get the timer hour.
- void setTimer (bool stat)

  Set the Timer object to set the timer status (ON/OFF)
- void add1s ()

  Add 1 secound to timer.
- void add5s ()

  Add 5 secounds to timer.
- void add10s ()

  Add 10 secounds to timer.
- void add1m ()

  Add 1 minute to timer.
- void add5m ()

  Add 5 minutes to timer.
- void add10m ()

  Add 10 minutes to timer.
- void rem1s ()

  Remove 1 secound from timer.
- void rem5s ()

  Remove 5 secounds from timer.
- void rem10s ()

  Remove 10 secounds from timer.
- void rem1m ()

  Remove 1 minute from timer.
- void rem5m ()

  Remove 5 minutes from timer.
- void rem10m ()

  Remove 10 minutes from timer.
- void resetTimer ()

  Reset timer.

Variables

- bool flagTimerStatus = OFF

  Timer Functions.
- const unsigned long updateInterval = 1000

2.13.1 Function Documentation

2.13.1.1 add10m()

```
void add10m ()
```

Add 10 minutes to timer.

Add 5 minutes to timer. Here is the call graph for this function: Here is the caller graph for this function:

2.13.1.2 add10s()

```
void add10s ()
```

Add 10 secounds to timer.

Add 5 secounds to timer. Here is the call graph for this function: Here is the caller graph for this function:

2.13.1.3 add1m()

```
void add1m ()
```

Add 1 minute to timer.

Add 10 secounds to timer. Here is the call graph for this function: Here is the caller graph for this function:

2.13.1.4 add1s()

```
void add1s ()
```

Add 1 secound to timer.

Get timer status. Here is the call graph for this function: Here is the caller graph for this function:

2.13.1.5 add5m()

```
void add5m ()
```

Add 5 minutes to timer.

Add 1 minute to timer. Here is the call graph for this function: Here is the caller graph for this function:

2.13.1.6 add5s()

```
void add5s ()
```

Add 5 secounds to timer.

Add 1 secound to timer. Here is the call graph for this function: Here is the caller graph for this function:

2.13.1.7 getTimerHour()

```
int getTimerHour ()
```

Get the Timer Hour object to get the timer hour.

Get timer minute.

Returns

int

Return the timer hours left in the timerHere is the call graph for this function: Here is the caller graph for this function:

2.13.1.8 getTimerMinute()

```
int getTimerMinute ()
```

Get the Timer Minute object to get the timer minute.

Get timer secound.

Returns

int

Return the timer minutes left in the timerHere is the call graph for this function: Here is the caller graph for this function:

2.13.1.9 getTimerSecound()

```
int getTimerSecound ()
```

Get the Timer Secound object to get the timer secound.

Timer for web operations.

Returns

int

Return the timer seconds left in the timer
Here is the call graph for this function: Here is the caller graph for this function:

2.13.1.10 getTimerStatus()

```
bool getTimerStatus ()
```

Get the Timer Status object to get the timer status (ON/OFF)

Get timer hour.

Note

This function is called by the processor function in AP_Destiller.cpp

Returns

true

false

Return the timer status (ON/OFF)Here is the caller graph for this function:

2.13.1.11 rem10m()

```
void rem10m ()
```

Remove 10 minutes from timer.

Remove 5 minutes from timer. Here is the call graph for this function: Here is the caller graph for this function:

2.13.1.12 rem10s()

```
void rem10s ()
```

Remove 10 secounds from timer.

Remove 5 secounds from timer. Here is the call graph for this function: Here is the caller graph for this function:

2.13.1.13 rem1m()

```
void rem1m ()
```

Remove 1 minute from timer.

Remove 10 secounds from timer. Here is the call graph for this function: Here is the caller graph for this function:

2.13.1.14 rem1s()

```
void rem1s ()
```

Remove 1 secound from timer.

Add 10 minutes to timer. Here is the call graph for this function: Here is the caller graph for this function:

2.13.1.15 rem5m()

```
void rem5m ()
```

Remove 5 minutes from timer.

Remove 1 minute from timer. Here is the call graph for this function: Here is the caller graph for this function:

2.13.1.16 rem5s()

```
void rem5s ()
```

Remove 5 secounds from timer.

Remove 1 secound from timer. Here is the call graph for this function: Here is the caller graph for this function:

2.13.1.17 resetTimer()

```
void resetTimer ()
```

Reset timer.

Remove 10 minutes from timer. Here is the call graph for this function: Here is the caller graph for this function:

2.13.1.18 sendTimer()

```
void sendTimer ()
```

Send timer to web.

This function sends the timer to the web interface to update the timer display in the web interface.

Note

> This function is called by the webTimer function in MD_Timer.cpp

Old values to compare with the new values to send updates only when the timer changes

Only send updates if the timer has changed

Send the timer to the web interface

Create the message to send to the web interface as a JSON object

Send the message to the web interface

Update the old valuesHere is the call graph for this function: Here is the caller graph for this function:

2.13.1.19 setTimer()

```
void setTimer (
            bool stat)
```

Set the Timer object to set the timer status (ON/OFF)

Reset timer.

Parameters

stat	

Set the timer status (ON/OFF)Here is the call graph for this function: Here is the caller graph for this function:

2.13.1.20 webTimer()

```
long webTimer (
            String op,
            int16_t amount)
```

Web timer function to manage the timer operations in the web interface.

Timer Functions calls.

This function is responsible for managing the timer operations.

Parameters

op	
amount	

Returns

unsigned long

Timer operations to add, remove or reset the timer value in miliseconds

Add time to the timer

Add 1 second

Add 5 seconds

Add 10 seconds

Add 1 minute

Add 5 minutes

Add 10 minutes

Remove time from the timer

Remove 1 second

Remove 5 seconds

Remove 10 seconds

Remove 1 minute

Remove 5 minutes

Remove 10 minutes

Reset the timer to 0

Timer management to decrease the timer every second and turn it off when it reaches 0 seconds left

Decrease the timer every second if the timer is ON

Turn off the timer when it reaches 0 seconds left

Only send updates at the defined interval

Send the timer to the web interfaceHere is the call graph for this function: Here is the caller graph for this function:

2.13.2 Variable Documentation

2.13.2.1 flagTimerStatus

```
bool flagTimerStatus = OFF
```

Timer Functions.

2.13.2.2 updateInterval

```
const unsigned long updateInterval = 1000
```

2.14 C:/workspace/Projeto/InteligentDestiller/src/MD_Web_Handlers.cpp File Reference

Web Handlers for the MD project.

```
#include "MD.h"
```

Include dependency graph for MD_Web_Handlers.cpp:

Functions

- String handleTimer ()

  Definitions.
- String handleHour ()

  Handle Hour.
- String handleMinute ()

  Handle Minute.
- String handleSecound ()

  Handle Secound.
- String handleWifiQuality ()

  Handlers for Wifi Quality, SSID, TX and IP.
- String handleWifiSSID ()

  Handle Wifi SSID.
- String handleWifiTX ()

  Handle Wifi TX.
- String handleWifiIP ()

  Handle Wifi IP.
- String handleTimeAdjust (const String &adjustment)

  Timer Add/Remove Time.
- String handleTimerControl (const String &control)

  Timer Start/Stop/Reset.
- String handleState (const String &state)

  Handlers for Auto, Pump, WaterOut, Resistor, WaterIn and WaterMax With Response.
- String handleToggle (const String &toggle)

  Handlers for Toggle Auto, Pump, waterOut, Resistor, WaterIn and WaterMax.
- String processor (const String &var)

  Function to handle the HTTP requests for the variables.

2.14.1 Detailed Description

Web Handlers for the MD project.

Author

Sérgio Carmo

Version

1.0

2.14.2 Function Documentation

2.14.2.1 handleHour()

```
String handleHour ()
```

Handle Hour.

Handle timer.

Returns

String

Note

This function returns the timer hour

Here is the call graph for this function: Here is the caller graph for this function:

2.14.2.2 handleMinute()

```
String handleMinute ()
```

Handle Minute.

Handle hour.

Returns

String

Note

This function returns the timer minute

Here is the call graph for this function: Here is the caller graph for this function:

2.14.2.3 handleSecound()

```
String handleSecound ()
```

Handle Secound.

Handle minute.

Returns

String

Note

This function returns the timer secound

Here is the call graph for this function: Here is the caller graph for this function:

2.14.2.4 handleState()

```
String handleState (
            const String & state)
```

Handlers for Auto, Pump, WaterOut, Resistor, WaterIn and WaterMax With Response.

Handle timer control.

Handle State

Parameters

state	

Returns

String

Note

This function handles the state of the variable (auto, manual, pump, waterOut, resistor, waterIn, waterMax, waterMin, waterAlarm)

This function returns the state of the variable (auto, manual, pump, waterOut, resistor, waterIn, waterMax, waterMin, waterAlarm)

Get the state of the variable (auto, manual, pump, waterOut, resistor, waterIn, waterMax, waterMin, waterAlarm)

Get the auto mode indicator state

Get the manual mode indicator state

Get the pump state

Get the water out valve state

Get the resistor state

Get the water in valve state

Get the water max state

Get the water min state

Get the water alarm state

Return nothing if the state is not foundHere is the call graph for this function: Here is the caller graph for this function:

2.14.2.5 handleTimeAdjust()

```
String handleTimeAdjust (
            const String & adjustment)
```

Timer Add/Remove Time.

Handle wifi IP.

Handle Time Adjust

Parameters

adjustment	

Returns

String

Note

This function adjusts the timer

This function returns the adjustment message for the timer (add1s, add5s, add10s, add1m, add5m, add10m, rem1s, rem5s, rem10s, rem1m, rem5m, rem10m)

Adjust the timer based on the adjustment parameter

Add 1 secound to timer

Add 5 secounds to timer

Add 10 secounds to timer

Add 1 minute to timer

Add 5 minutes to timer

Add 10 minutes to timer

Remove 1 secound from timer

Remove 5 secounds from timer

Remove 10 secounds from timer

Remove 1 minute from timer

Remove 5 minutes from timer

Remove 10 minutes from timer

Return the adjustment message for the timer (add1s, add5s, add10s, add1m, add5m, add10m, rem1s, rem5s, rem10s, rem1m, rem5m, rem10m)Here is the call graph for this function: Here is the caller graph for this function:

2.14.2.6 handleTimer()

```
String handleTimer ()
```

Definitions.

Get resistor.

Timer handlers

Handle Timer

Returns

String

Note

This function handles the timer

This function returns the timer in the format HH:MM:SS

Here is the call graph for this function: Here is the caller graph for this function:

2.14.2.7 handleTimerControl()

```
String handleTimerControl (
            const String & control)
```

Timer Start/Stop/Reset.

Handle time adjustment.

Handle Timer Control

Parameters

control	

Returns

String

Note

This function controls the timer

This function returns the control message for the timer (start, stop, reset)

Control the timer based on the control parameter (start, stop, reset)

Start the timer

Stop the timer

Reset the timer

Return the control message for the timer (start, stop, reset)Here is the call graph for this function: Here is the caller graph for this function:

2.14.2.8 handleToggle()

```
String handleToggle (
            const String & toggle)
```

Handlers for Toggle Auto, Pump, waterOut, Resistor, WaterIn and WaterMax.

Handle state.

Handle Toggle

Parameters

toggle	

Returns

String

Note

This function toggles the variable (auto, pump, waterOut, resistor, waterIn)

This function returns the toggle message for the variable (auto, pump, waterOut, resistor, waterIn)

Toggle the variable (auto, pump, waterOut, resistor, waterIn)

Toggle the auto mode indicator state

Toggle the pump state

Toggle the water out valve state

Toggle the water in valve state

Toggle the resistor state

Return the toggle message for the variable (auto, pump, waterOut, resistor, waterIn)Here is the call graph for this function: Here is the caller graph for this function:

2.14.2.9 handleWifiIP()

```
String handleWifiIP ()
```

Handle Wifi IP.

Handle wifi TX.

Returns

String

Note

This function returns the WiFi IP address

Here is the caller graph for this function:

2.14.2.10 handleWifiQuality()

```
String handleWifiQuality ()
```

Handlers for Wifi Quality, SSID, TX and IP.

Handle secound.

Handle Wifi Quality

Returns

String

Note

This function returns the WiFi quality in percentage

Here is the call graph for this function: Here is the caller graph for this function:

2.14.2.11 handleWifiSSID()

```
String handleWifiSSID ()
```

Handle Wifi SSID.

Handle wifi quality.

Returns

String

Note

This function returns the WiFi SSID

Here is the caller graph for this function:

2.14.2.12 handleWifiTX()

```
String handleWifiTX ()
```

Handle Wifi TX.

Handle wifi SSID.

Returns

String

Note

This function returns the WiFi TX power

Here is the caller graph for this function:

2.14.2.13 processor()

```
String processor (
             const String & var)
```

Function to handle the HTTP requests for the variables.

Handle toggle.

Processor

Parameters

var	

Returns

String

Note

This function processes the variables for the web server requests

This function is called by the web server handle function (server.on)

This function returns the variable value for the web server request

Process the variable for the web server request (var) and return the variable value

Timer variable request

Return the timer value in the format HH:MM:SS

WaterMax variable request

Return the waterMax value

WaterMin variable request

Return the waterMin value

WaterAlarm variable request

Return the waterAlarm value

Auto variable request

Return the auto value

Manual variable request

Return the manual value

Pump variable request

Return the pump value

WaterOut variable request

Return the waterOut value

Resistor variable request

Return the resistor value

WaterIn variable request

Return the waterIn value

WifiQuality variable request

Return the wifiQuality value

WifiSSID variable request

Return the wifiSSID value

WifiTX variable request

Return the wifiTX value

WifiIP variable request

Return the wifiIP value

TimerHour variable request

Return the timerHour value

TimerMinute variable request

Return the timerMinute value

TimerSecound variable request

Return the timerSecound value

Return nothing if the variable is not foundHere is the call graph for this function: Here is the caller graph for this function:

2.15 C:/workspace/Projeto/InteligentDestiller/src/MD_Web_IO_↩ Mngmnt.cpp File Reference

```
#include "MD.h"
```
Include dependency graph for MD_Web_IO_Mngmnt.cpp:

Functions

- void updateOutputsWeb ()

  Output variables image.
- bool getAutoModeWeb ()

  GETS BLOCK.
- bool getResistorWeb ()

  Get the Ind Min object.
- bool getValv_Water_InWeb ()

  Get the Ind Alarm object.
- bool getValv_Water_OutWeb ()

  Get the Valv Water Out object.
- bool getPumpWeb ()

  Get the Pump object.
- void setAutoModeWeb (bool state)

SETS BLOCK.
- void setPumpWeb (bool state)

  Set the Pump object.
- void setValveWaterInWeb (bool state)

  Set the Valve Water In object.
- void setValveWaterOutWeb (bool state)

  Set the Valve Water Out object.
- void setResistorWeb (bool state)

  Set the Resistor object.
- void toggleAutoModeWeb ()

  TOGGLES BLOCK.
- void togglePumpWeb ()

  Toggle Pump Web.
- void toggleValveWaterInWeb ()

  Toggle Valve Water In Web.
- void toggleValveWaterOutWeb ()

  Toggle Valve Water Out Web.
- void toggleResistorWeb ()

  Toggle Resistor Web.

Variables

- uint8_t outputVarsImageWeb [9]

2.15.1 Detailed Description

Author

Sérgio Carmo (19749@ipportalegre.pt)

Version

0.1

2.15.2 Function Documentation

2.15.2.1 getAutoModeWeb()

```
bool getAutoModeWeb ()
```

GETS BLOCK.

Set resistor.

Get the Ind Max object

This function returns the state of the max indicator.

Returns

true

false

Here is the caller graph for this function:

2.15.2.2 getPumpWeb()

```
bool getPumpWeb ()
```

Get the Pump object.

Get auto mode.

This function returns the state of the water pump.

Returns

true

false

Here is the caller graph for this function:

2.15.2.3 getResistorWeb()

```
bool getResistorWeb ()
```

Get the Ind Min object.

Get valve water out.

This function returns the state of the min indicator.

Returns

true

false

Here is the caller graph for this function:

2.15.2.4 getValv_Water_InWeb()

```
bool getValv_Water_InWeb ()
```

Get the Ind Alarm object.

Get pump state.

This function returns the state of the alarm indicator.

Returns

true

false

Here is the caller graph for this function:

2.15.2.5 getValv_Water_OutWeb()

```
bool getValv_Water_OutWeb ()
```

Get the Valv Water Out object.

Get valve water in.

This function returns the state of the water out valve.

Returns

true

false

Here is the caller graph for this function:

2.15.2.6 setAutoModeWeb()

```
void setAutoModeWeb (
            bool state)
```

SETS BLOCK.

Toggle resistor.

Set the Auto Mode object

This function sets the state of the auto mode.

Parameters

state	

Returns

void

Static variable to store the old state

Set the auto indicator to ON

Set the old state to the current state

Set the auto indicator to OFF

Set the old state to the current stateHere is the caller graph for this function:

2.15.2.7 setPumpWeb()

```
void setPumpWeb (
            bool state)
```

Set the Pump object.

Set auto mode.

This function sets the state of the pump.

Parameters

state	

Returns

void

Static variable to store the old state

Set the pump to ON

Set the old state to the current state

Set the pump to OFF

Set the old state to the current stateHere is the caller graph for this function:

2.15.2.8 setResistorWeb()

```
void setResistorWeb (
            bool state)
```

Set the Resistor object.

Set valve water out.

This function sets the state of the resistor.

Parameters

state	

Returns

void

Static variable to store the old state

Set the resistor to ON

Set the old state to the current state

Set the resistor to OFF

Set the old state to the current stateHere is the caller graph for this function:

2.15.2.9 setValveWaterInWeb()

```
void setValveWaterInWeb (
            bool state)
```

Set the Valve Water In object.

Set pump state.

This function sets the state of the water in valve.

Parameters

state	

Returns

void

Static variable to store the old state

Set the water in valve to ON

Set the old state to the current state

Set the water in valve to OFF

Set the old state to the current stateHere is the caller graph for this function:

2.15.2.10 setValveWaterOutWeb()

```
void setValveWaterOutWeb (
             bool state)
```

Set the Valve Water Out object.

Set valve water in.

This function sets the state of the water out valve.

Parameters

state	

Returns

void

Static variable to store the old state

Set the water out valve to ON

Set the old state to the current state

Set the water out valve to OFF

Set the old state to the current stateHere is the caller graph for this function:

2.15.2.11 toggleAutoModeWeb()

```
void toggleAutoModeWeb ()
```

TOGGLES BLOCK.

Update outputs for the web interface.

Toggle Auto Mode Web

This function toggles the auto mode.

Returns

void

Get the auto indicator state

Toggle the auto indicator state

Set the auto indicator state

Toggle the auto indicator state

Set the auto indicator stateHere is the call graph for this function: Here is the caller graph for this function:

2.15.2.12 togglePumpWeb()

```
void togglePumpWeb ()
```

Toggle Pump Web.

Toggle auto mode.

This function toggles the pump.

Returns

void Toggle Pump

Get the pump state

Toggle the pump state

Set the pump state

Print the pump state

Print the pump state

Toggle the pump state

Set the pump state

Print the pump state

Print the pump stateHere is the call graph for this function: Here is the caller graph for this function:

2.15.2.13 toggleResistorWeb()

```
void toggleResistorWeb ()
```

Toggle Resistor Web.

Toggle valve water out.

This function toggles the resistor.

Returns

void Toggle Resistor

Get the resistor state

Toggle the resistor state

Set the resistor state

Toggle the resistor state

Set the resistor stateHere is the call graph for this function: Here is the caller graph for this function:

2.15.2.14 toggleValveWaterInWeb()

```
void toggleValveWaterInWeb ()
```

Toggle Valve Water In Web.

Toggle pump state.

This function toggles the water in valve.

Returns

void Toggle Valve Water In

Get the water in valve state

Toggle the water in valve state

Set the water in valve state

Toggle the water in valve state

Set the water in valve stateHere is the call graph for this function: Here is the caller graph for this function:

2.15.2.15 toggleValveWaterOutWeb()

```
void toggleValveWaterOutWeb ()
```

Toggle Valve Water Out Web.

Toggle valve water in.

This function toggles the water out valve.

Returns

void Toggle Valve Water Out

Get the water out valve state

If the water out valve state is OFF and the manual mode is ON

Toggle the water out valve state

Set the water out valve state

Toggle the water out valve state

Set the water out valve stateHere is the call graph for this function: Here is the caller graph for this function:

2.15.2.16 updateOutputsWeb()

```
void updateOutputsWeb ()
```

Output variables image.

Mode management.

FUNCTIONS BLOCK

Update Outputs Web

This function updates the outputs for the web interface.

Returns

void

Update the resistor state

Update the pump state

Update the water out valve state

Update the water in valve state

Update the alarm state

Update the water min state

Update the water max state

Update the manual mode state

outputVarsImageWeb[7] = getIndAuto(); /// Update the auto indicator stateHere is the call graph for this function: Here is the caller graph for this function:

2.15.3 Variable Documentation

2.15.3.1 outputVarsImageWeb

```
uint8_t outputVarsImageWeb[9]
```

2.16 C:/workspace/Projeto/InteligentDestiller/src/MD_wifi.cpp File Reference

```
#include "MD.h"
```

Include dependency graph for MD_wifi.cpp:

Functions

- void OnWiFiEvent (WiFiEvent_t event)

  WiFi Event Handler.
- String wifiQuality ()

  WiFi Quality.
- void connectToWIFI ()

  Connect to WiFi.
- void connectToSoftAP ()

  Connect to SoftAP.
- bool initWIFI ()

  Initiate WiFi.

Variables

- const char $*$ soft_ap_ssid = "Destiller-AP"
- const char $*$ soft_ap_password = "Destiller-AP"

2.16.1 Function Documentation

2.16.1.1 connectToSoftAP()

```
void connectToSoftAP ()
```

Connect to SoftAP.

Returns

void

Note

> This function creates a Soft Access Point
>
> The IP Address of the ESP32 Soft Access Point is set to: 192.168.100.100

Set the hostname for the Soft Access Point

Set the WiFi event handler

Enable auto reconnect to the WiFi network

Set to Station + Access Point mode

Start the Soft Access Point with the SSID and password

Get the handle for the Soft Access Point interface

Define IP configuration structure and set desired IPs

IP configuration structure

AP IP

Gateway IP

Subnet Mask

Set the IP configuration to the soft AP interface

Stop DHCP server to change settings

Apply new IP settings

Restart DHCP server with new settings

Print the hostname

Print the IP address of the ESP32 Soft Access Point

Print the IP address of the ESP32 Soft Access PointHere is the call graph for this function: Here is the caller graph for this function:

2.16.1.2 connectToWIFI()

```
void connectToWIFI ()
```

Connect to WiFi.

Returns

void

Note

This function connects the ESP32 to a WiFi network

End the server

Dark theme

Timeout to web server

Timeout to connect

Debug output

Hostname

Hostname

Set the WiFiManager parameters and connect to the WiFi network with the SSID and password stored in the EEPROM memory

Get the SSID and password from the WiFiManager and connect to the WiFi network

Get the SSID from the WiFiManager

Get the password from the WiFiManager

Connect to the WiFi network with the SSID and password

Connect to the WiFi network with the SSID and password

Delay for 10 ms

Wait for the ESP32 to connect to the WiFi network and print the connection statusHere is the call graph for this function: Here is the caller graph for this function:

2.16.1.3 initWIFI()

```
bool initWIFI ()
```

Initiate WiFi.

Returns

bool

Note

This function initiates the WiFi connection

SoftAP flag to create a Soft Access Point only once

WIFI_MODE_OPTIONS

1 - Connect to local WiFi network only

2 - Create a local AP (SoftAP)

3 - Both (Connect to local WiFi network and create a local AP)

Initiate WiFi

Initiate WiFi in AP mode only

Initiate WiFi

Initiate WiFi in AP modeHere is the call graph for this function: Here is the caller graph for this function:

2.16.1.4 OnWiFiEvent()

```
void OnWiFiEvent (
            WiFiEvent_t event)
```

WiFi Event Handler.

Note

This function handles the WiFi events

Parameters

event	

Returns

void

Handle WiFi events

Station connected to WiFi network

Soft Access Point started

Station connected to ESP32 Soft Access Point

Station disconnected from ESP32 Soft Access PointHere is the call graph for this function: Here is the caller graph for this function:

2.16.1.5 wifiQuality()

```
String wifiQuality ()
```

WiFi Quality.

Inicia o wifi.

Returns

String

Note

This function returns the WiFi quality in percentage

Get the Received Signal Strength Indicator (RSSI) in dBm

Quality in percentage

Quality in string format

Last quality in percentage

Last RSSI in dBm

If the last RSSI is equal to the current RSSI

Return the quality in string format with RSSI and quality in percentage values

Calculate the quality in percentage

If the RSSI is less than or equal to -100 dBm

Set the quality to 0%

If the RSSI is greater than or equal to -5 dBm

Set the quality to 100%

If the RSSI is between -100 dBm and -5 dBm

Calculate the quality in percentage

Return the quality in string format with RSSI and quality in percentage values

Send the quality in percentage to the web server as a JSON object

Return the quality in string format with RSSI and quality in percentage values

Update the last RSSI with the current RSSIHere is the caller graph for this function:

2.16.2 Variable Documentation

2.16.2.1 soft_ap_password

```
const char* soft_ap_password = "Destiller-AP"
```

2.16.2.2 soft_ap_ssid

```
const char* soft_ap_ssid = "Destiller-AP"
```

2.17 C:/workspace/Projeto/InteligentDestiller/src/OS_FS.cpp File Reference

File System functions.

```
#include "OS.h"
```

Include dependency graph for OS_FS.cpp:

Functions

- void initFS ()
  Initialize Serial Terminal.

2.17.1 Detailed Description

File System functions.

Author

Sérgio Carmo (19749@ipportalegre.pt)

Version

0.1

2.17.2 Function Documentation

2.17.2.1 initFS()

```
void initFS ()
```

Initialize Serial Terminal.

Inicia o segundo core.

Initialize File System (SPIFFS)

This function is responsible for initializing the File System (SPIFFS).

Note

Initialize SPIFFS

Returns

void

```
initFS();
```

Initialize SPIFFSHere is the call graph for this function: Here is the caller graph for this function:

2.18 C:/workspace/Projeto/InteligentDestiller/src/OS_GPIO.cpp File Reference

GPIO functions.

```
#include "OS.h"
```

Include dependency graph for OS_GPIO.cpp:

Functions

- void readInputs ()

  Alarm state.
- void writeOutputs ()

  Write Outputs from array image to GPIO.
- void initPinsInputs ()

  Configure GPIOs as INPUT.
- void initPinsOutputs ()

  Configure GPIOs as OUTPUTs.
- bool getWaterMax ()

  GETS BLOCK.
- bool getWaterMin ()

  Get the Water Min object.
- bool getAlarm ()

  Get the Alarm object.
- bool getAlarmIND ()

  Get the Alarm State object.
- bool getAutoModeSW ()

  Get the Auto Mode SW object.
- bool getManualMode ()

  Get the Manual Mode object.
- bool getIndAuto ()

  Get the Ind Max object.
- bool getResistor ()

  Get the Ind Min object.
- bool getValv_Water_In ()

  Get the Ind Alarm object.
- bool getValv_Water_Out ()

  Get the Valv Water Out object.
- bool getPump ()

  Get the Pump object.
- void setIndMax (bool state)

  SETS BLOCK.
- void setIndMin (bool state)

  Set the Ind Min object.
- void setIndAlarm (bool state)

  Set the Ind Alarm object.
- void setIndMan (bool state)

  Set the Ind Man object.
- void setIndAuto (bool state)

  Set the Auto Mode object.

- void setPump (bool state)

    Set the Pump object.
- void setValveWaterIn (bool state)

    Set the Valve Water In object.
- void setValveWaterOut (bool state)

    Set the Valve Water Out object.
- void setResistor (bool state)

    Set the Resistor object.

Variables

- uint8_t inputVarsImage [5]

    GPIO Images.
- const uint8_t inputVarsPINs [ ] = {PIN_SW_MAN, PIN_SMIN, PIN_SMAX, PIN_SW_AUTO, PIN_SALARM}

    Input variables image.
- uint8_t outputVarsImage [9]

    Input variables pins.
- const uint8_t outputVarsPINs [ ]

    Output variables image.
- bool ALARM_STATE = OFF

    Output variables pins.

2.18.1 Detailed Description

GPIO functions.

Author

Sérgio Carmo (19749@ipportalegre.pt)

Version

0.1

2.18.2 Function Documentation

2.18.2.1 getAlarm()

```
bool getAlarm ()
```

Get the Alarm object.

Retorna estado do sensor de agua de minimo.

This function returns the state of the alarm.

Returns

true

false

Here is the caller graph for this function:

2.18.2.2 getAlarmIND()

```
bool getAlarmIND ()
```

Get the Alarm State object.

Retorna estado do alarme.

This function returns the state of the alarm.

Returns

true

false

Here is the caller graph for this function:

2.18.2.3 getAutoModeSW()

```
bool getAutoModeSW ()
```

Get the Auto Mode SW object.

Retorna estado do modo auto.

This function returns the state of the auto mode switch.

Returns

true

false

Here is the caller graph for this function:

2.18.2.4 getIndAuto()

```
bool getIndAuto ()
```

Get the Ind Max object.

Retorna estado do indicador de alarme.

This function returns the state of the max indicator.

Returns

true

false

Here is the caller graph for this function:

2.18.2.5 getManualMode()

```
bool getManualMode ()
```

Get the Manual Mode object.

Retorna estado do interruptor auto.

This function returns the state of the manual mode switch.

Returns

true

false

Here is the caller graph for this function:

2.18.2.6 getPump()

```
bool getPump ()
```

Get the Pump object.

Retorna estado da valvula de vapor.

This function returns the state of the water pump.

Returns

true

false

Here is the caller graph for this function:

2.18.2.7 getResistor()

```
bool getResistor ()
```

Get the Ind Min object.

Retorna estado do botao manual.

This function returns the state of the min indicator.

Returns

true

false

Here is the caller graph for this function:

2.18.2.8 getValv_Water_In()

```
bool getValv_Water_In ()
```

Get the Ind Alarm object.

Retorna estado do resistor.

This function returns the state of the alarm indicator.

Returns

true

false

Here is the caller graph for this function:

2.18.2.9 getValv_Water_Out()

```
bool getValv_Water_Out ()
```

Get the Valv Water Out object.

Retorna estado da entrada de agua.

This function returns the state of the water out valve.

Returns

true

false

Here is the caller graph for this function:

2.18.2.10 getWaterMax()

```
bool getWaterMax ()
```

GETS BLOCK.

Inicia os inputs.

Get the Water Max object

This function returns the state of the water max sensor.

Returns

true

false

Here is the caller graph for this function:

2.18.2.11 getWaterMin()

```
bool getWaterMin ()
```

Get the Water Min object.

Retorna estado do sensor de agua de maximo.

This function returns the state of the water min sensor.

Returns

true

false

Here is the caller graph for this function:

2.18.2.12 initPinsInputs()

```
void initPinsInputs ()
```

Configure GPIOs as INPUT.

Inicia os outputs.

Configure GPIOs as INPUTs

This function configures the GPIOs as INPUTs. The GPIOs are used to read the state of the switches and sensors. on/off sw

min sensor

max sensor

push sw

alarmHere is the caller graph for this function:

2.18.2.13 initPinsOutputs()

```
void initPinsOutputs ()
```

Configure GPIOs as OUTPUTs.

Inicia o sistema de arquivos.

Configure GPIOs as OUTPUTs

This function configures the GPIOs as OUTPUTs. The GPIOs are used to control the indicators, valves, pump, and resistor. resistor

water in valve

water out valve

water pump

min indicator

max indicator

alarm indicator

manual indicator

auto indicatorHere is the caller graph for this function:

2.18.2.14 readInputs()

```
void readInputs ()
```

Alarm state.

Atualiza as saidas.

Read ESP32 GPIOs and stores in the image array

Read all inputs and store in the image array

This function reads all the inputs and stores them in the image array. The image array is used to store the current state of the inputs and outputs. PIN_INPUT_COUNT = 5

Read the input from the GPIOHere is the caller graph for this function:

2.18.2.15 setIndAlarm()

```
void setIndAlarm (
            bool state)
```

Set the Ind Alarm object.

Liga/desliga o indicador luminoso nivel agua minimo.

This function sets the state of the alarm indicator.

Parameters

state	

Returns

void

Old state of the indicator

Condition to debugHere is the call graph for this function: Here is the caller graph for this function:

2.18.2.16 setIndAuto()

```
void setIndAuto (
            bool state)
```

Set the Auto Mode object.

Liga/desliga o indicador luminoso da resistencia.

This function sets the state of the auto mode.

Parameters

state	

Returns

void

Here is the call graph for this function: Here is the caller graph for this function:

2.18.2.17 setIndMan()

```
void setIndMan (
            bool state)
```

Set the Ind Man object.

Liga/desliga o indicador luminoso alarme.

This function sets the state of the manual indicator.

Parameters

state	

Returns

void

Here is the call graph for this function: Here is the caller graph for this function:

2.18.2.18 setIndMax()

```
void setIndMax (
            bool state)
```

SETS BLOCK.

Retorna estado da bomba de agua.

Set the Ind Max object

This function sets the state of the max indicator.

Parameters

state	

Returns

void

Old state of the indicator

Set the max indicator to ON

Condition to debug

Send the max indicator value to the web server

Set the max indicator to OFF

Send the max indicator value to the web serverHere is the call graph for this function: Here is the caller graph for this function:

2.18.2.19 setIndMin()

```
void setIndMin (
            bool state)
```

Set the Ind Min object.

Liga/desliga o indicador luminoso nivel agua maximo.

This function sets the state of the min indicator.

Parameters

state	

Returns

void

Old state of the indicator

Set the min indicator to ON

Send the min indicator value to the web server

Set the min indicator to OFF

Send the min indicator value to the web serverHere is the call graph for this function: Here is the caller graph for this function:

2.18.2.20 setPump()

```
void setPump (
            bool state)
```

Set the Pump object.

Liga/desliga o indicador luminoso modo manual.

This function sets the state of the pump.

Parameters

state	

Returns

void

Here is the call graph for this function: Here is the caller graph for this function:

2.18.2.21 setResistor()

```
void setResistor (
            bool state)
```

Set the Resistor object.

Liga/desliga o indicador luminoso da valvula 2.

This function sets the state of the resistor.

Parameters

state	

Returns

void

Here is the call graph for this function: Here is the caller graph for this function:

2.18.2.22 setValveWaterIn()

```
void setValveWaterIn (
            bool state)
```

Set the Valve Water In object.

Liga/desliga o indicador luminoso da bomba de agua.

This function sets the state of the water in valve.

Parameters

state	

Returns

void

Here is the call graph for this function: Here is the caller graph for this function:

2.18.2.23 setValveWaterOut()

```
void setValveWaterOut (
            bool state)
```

Set the Valve Water Out object.

Liga/desliga o indicador luminoso da valvula 1.

This function sets the state of the water out valve.

Parameters

state	

Returns

void

Here is the call graph for this function: Here is the caller graph for this function:

2.18.2.24 writeOutputs()

```
void writeOutputs ()
```

Write Outputs from array image to GPIO.

Liga/desliga o modo auto.

Write all outputs from the image array to the GPIOs

This function writes all the outputs from the image array to the GPIOs. The image array is used to store the current state of the inputs and outputs. PIN_OUTPUT_COUNT = 9

Write the output to the GPIOHere is the caller graph for this function:

2.18.3 Variable Documentation

2.18.3.1 ALARM_STATE

```
bool ALARM_STATE = OFF
```

Output variables pins.

2.18.3.2 inputVarsImage

```
uint8_t inputVarsImage[5]
```

GPIO Images.

2.18.3.3 inputVarsPINs

```
const uint8_t inputVarsPINs[] = {PIN_SW_MAN, PIN_SMIN, PIN_SMAX, PIN_SW_AUTO, PIN_SALARM}
```

Input variables image.

2.18.3.4 outputVarsImage

```
uint8_t outputVarsImage[9]
```

Input variables pins.

2.18.3.5 outputVarsPINs

```
const uint8_t outputVarsPINs[]
```

Initial value:

```
= {PIN_RAQ, PIN_BMB, PIN_VALV_WATER_OUT, PIN_VALV_WATER_IN,
                                PIN_IND_ALARM, PIN_IND_MIN, PIN_IND_MAX, PIN_IND_AUTO, PIN_IND_MAN}
```

Output variables image.

2.19 C:/workspace/Projeto/InteligentDestiller/src/OS_multi_core.cpp File Reference

Operating System MultiCore functions.

```
#include "OS.h"
```

Include dependency graph for OS_multi_core.cpp:

Functions

- void initMultiCore ()

  Function to initiate the 2nd core to run 'loop2' function.

2.19.1 Detailed Description

Operating System MultiCore functions.

Author

Sérgio Carmo

Version

1.0

2.19.2 Function Documentation

2.19.2.1 initMultiCore()

```
void initMultiCore ()
```

Function to initiate the 2nd core to run 'loop2' function.

Main loop to run in core 1.

<Function to initiate the 2nd core> Implements the 2nd core option to run 'loop2' function

Function to implement the task

Name of the task

Stack size in words

Task input parameter

Priority of the task

Task handle.

Core where the task should runHere is the call graph for this function: Here is the caller graph for this function:

Index