

BIBLIOTECA — Spartan STABLETIME

A biblioteca de software Spartan STABLETIME faz parte de elementos do projeto de infraestrutura de automação SPARTA, e é baseado em C++ na versão específica para ESP8266, e requer a instalação de bibliotecas externas para a microcontroladora:

- WifiUdp.h
 - Créditos à equipe de colaboração pela criação da biblioteca utilizada: <https://github.com/arduino-libraries/>
- NTPClient.h
 - Créditos à equipe de colaboração pela criação da biblioteca utilizada: <https://github.com/arduino-libraries/>

Tem como função ser base para elaboração de projetos que necessitem acesso ao horário local, idealizada para geração de logs em infraestruturas de automação, com sistema de redundância em falhas de servidores de NTP ou conexões instáveis com os mesmos.

Até a presente data de 03/09/2022, está na versão 1.0, e está em estado de desenvolvimento pelo time de desenvolvimento responsável pelo desenvolvimento do protocolo de transmissão da camada de aplicação IGNU01, e tem usos que podem se relacionar ao citado software.

A composição da Classe pública principal engloba as seguintes funções:

```
bool begin();  
// Inicia a instância TIMESTABLE_SPARTAN  
// retorna TRUE se objeto foi inicializado com sucesso.  
  
void update();  
// atualiza a instância TIMESTABLE_SPARTAN  
// deve ser colocado no rotina de loop do código principal.  
  
String GetRunningTime();  
//Retorna valor STRING do tempo de funcionamento  
//da MicroControladora em montagem HH:MM:SS
```

```

String GetFormattedTIME();
String GetFormattedDATE();
//Retornam valores STRING na configuração
//"DD/MM/AAAA" ou "HH:MM:SS - Sem acréscimo do zero à
//esquerda do número em valores menores que dez;
//EX 10:09:05 = 10:9:5

int GetHours();
int GetMinute();
int GetSeconds();
int GetDay();
int GetMonth();
int GetYear();
// Retornam valores inteiros (INT) de acordo à função chamada.

```

Exemplo de uso:

```

#include "TIMESTABLE_SPARTAN.h"
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WiFiMulti.h>

TIMESTABLE_SPARTAN TIMESTABLE_SPARTAN;

int i = 0;

void setup() {

  Serial.begin(115200);
  Serial.println(" ");
  Serial.println("Starting_System...");

  WiFi.mode(WIFI_STA);
  WiFi.hostname("SPARTAN_TEST_TIMESTABLE");
  ESP8266WiFiMulti wifiMultiSSID;

  wifiMultiSSID.addAP("1st_Wifi_Network", "wifi_password");
  wifiMultiSSID.addAP("2nd_Wifi_Network", "wifi_password");

```

```

if (wifiMultiSSID.run() == WL_CONNECTED)
{
    Serial.print("Connected to ");
    Serial.println(WiFi.SSID());
    Serial.print("IP address: ");
    Serial.println(WiFi.localIP());

    if(TIMESTABLE_SPARTAN.begin()){

        Serial.println("Started_Succesfully!");

    }
}

void loop() {
    TIMESTABLE_SPARTAN.update();

    if (i == 5000){ // merely for demonstration porpuses;
        Serial.println("\n\nSystem Time:");
        Serial.println(TIMESTABLE_SPARTAN.GetRunningTime());

        Serial.println("Current Time:");
        Serial.println(TIMESTABLE_SPARTAN.GetFormatedTIME());

        Serial.println("Current Date:");
        Serial.println(TIMESTABLE_SPARTAN.GetFormatedDATE());

        i=0;
    }
    i++;
    delay (1);
}

```

Resultado esperado:

```

System Time:
0:50:59
Current Time:
20:49:49
Current Date:
3/9/2022

```