



# Projektová dokumentace

## Měření teploty

IMP - Mikroprocesorové a vestavěné systémy

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Zapojení a nastavení hardware</b>	<b>1</b>
2.1	Zapojení teplotního čidla . . . . .	1
2.2	Zapojení LED diody . . . . .	1
<b>3</b>	<b>Řešení projektu</b>	<b>2</b>
3.1	GPIO . . . . .	2
3.2	Úložiště . . . . .	2
3.3	Wifi . . . . .	3
3.4	Webserver . . . . .	3
3.5	Měření teploty . . . . .	3
3.6	Synchronizace s NTP serverem . . . . .	3
3.7	LED Dioda . . . . .	4
3.8	Hlavní smyčka . . . . .	4
<b>4</b>	<b>Testování</b>	<b>4</b>
<b>5</b>	<b>Závěr</b>	<b>4</b>
<b>6</b>	<b>Videa</b>	<b>4</b>

# 1 Úvod

Cílem projektu bylo vytvořit aplikaci pro zařízení ESP32, které dokáže měřit teplotu pomocí čidla LMT85LPG<sup>1</sup>, hodnoty ukládat s časovou značkou RTC, synchronizací s NTP serverem a přístupem k ovládání přes HTTP server. Použitý senzor poskytuje analogové údaje prostřednictvím hodnoty napětí v mV na jeho výstupu.

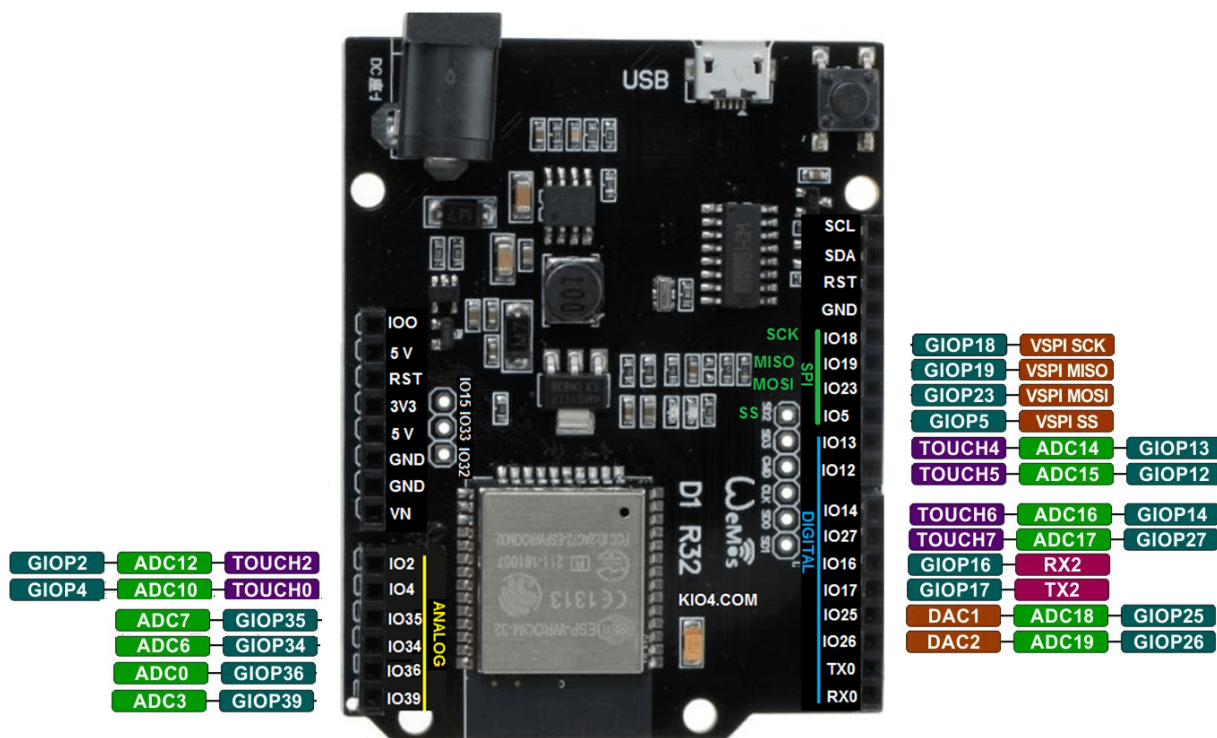
## 2 Zapojení a nastavení hardware

### 2.1 Zapojení teplotního čidla

Teplotní čidlo obsahuje 3 konektory - napájení (3.3V), zem a výstupní analogový signál. Napájení bylo připojeno pomocí nepájivého pole na výstup ESP 3.3V a zem, pro data byl použit vstup ESP GPIO36/ADC6 viz obrázek 1.

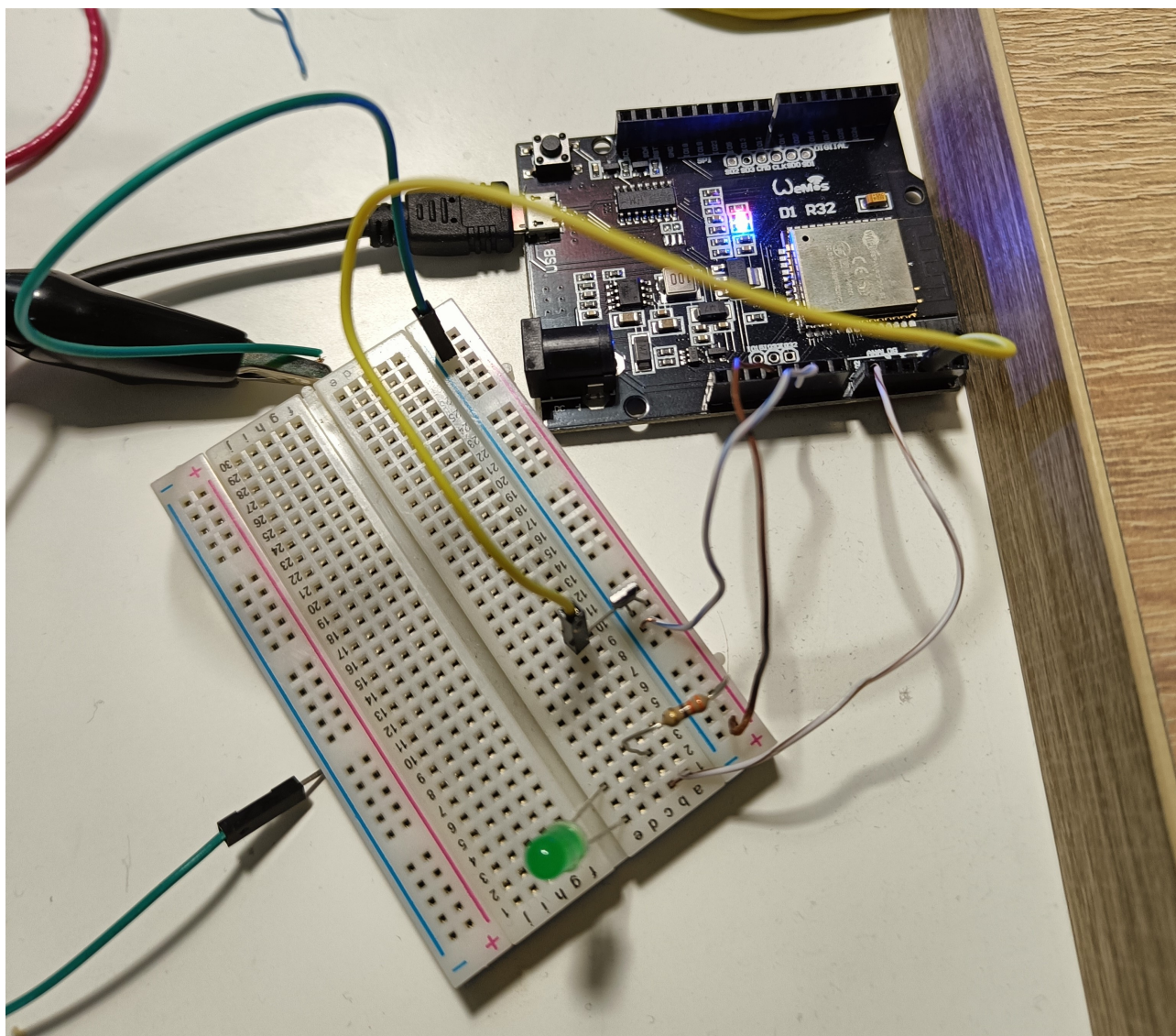
### 2.2 Zapojení LED diody

Dioda obsahuje 2 konektory, katodu a anodu, na katodu jsem připojil předřazený odpor pro snížení vstupního napětí na 2V namísto 3.3V. Katodu jsem připojil k ESP GPIO36 nakonfigurovaný jako výstup. Pro vypnutí diody je výstup v logické 1, pro rozsvícení naopak v logické 0.



Obrázek 1: Schéma desky

<sup>1</sup><https://www.ti.com/lit/ds/symlink/lmt85.pdf>



Obrázek 2: Zapojení komponent

### 3 Řešení projektu

Aplikace je řešena pomocí IDF a Platformio.

#### 3.1 GPIO

Jako první po o spuštění samotného ESP, aplikace inicializuje GPIO, nastavením pinu pro LED diodu na mód OUTPUT a vypnutí PULLUP a PULLDOWN. Informaci o který GPIO pin se jedná můžeme změnit v `#define LED_PIN`.

#### 3.2 Úložiště

Následně se začne inicializovat lokální úložiště pro ukládání historie měření. Jako úložiště se používá SPIFFS partition, nastavení pro platformio je uloženo v souboru `partition.csv`. Po inicializaci úložiště se provede kontrolní zápis do souboru, čímž se také vymaže veškerý jeho obsah. V případě, že chceme zachovat měření i přes vypnutí, stačí změnit funkci `save_string_to_file` na funkci

`append_string_to_file`, která namísto otevření s parametrem `read` otevře soubor s parametrem `append` a pouze přidá další řádek. Obě tyto funkce jsou naimplementovány pomocí funkcí `fopen`, `fprintf`, `fclose`. V případě neúspěchu se vypíše příslušná chybová hláška na `ESP_LOGE`.

### 3.3 Wifi

Další v pořadí inicializace je modul WiFi, který se pokusí připojit k nastavené síti ve zdrojovém kódu. Přístupové údaje k WiFi lze jednoduše změnit v souboru `main.c` pomocí deklarací `#define WIFI_SSID "název vaší wifi"` a `#define WIFI_PASS "heslo"`. Pro wifi je nutné inicializovat paměť NVS pomocí `nvs_flash_init()`. Inicializace wifi je implementována ve funkci `connect_wifi()`, kde se zavede event loop a vytvoří se handlers pro wifi a pro získání IP adresy. Každý handler má svoji implementaci v samostatné funkci. Následně se zkontroluje výsledek připojení porovnáním bitů a případě se vypíše chybová hláška. Pokud se nepodaří připojit k zadané síti, aplikace opakuje ještě 15 pokusů o připojení, pokud se i tak nepodaří k síti připojit aplikace končí. V případě úspěchu a získání IP adresy zařízení vypíše přidělenou IP adresu a nastartuje webserver pro připojení na získané IP adrese.

### 3.4 Webserver

Pokud se povede připojit k síti wifi, vytvoří se pomocí `xTaskCreate`<sup>2</sup> task pro provoz webserveru. Funkce přijímá jako handler funkci `http_server_task`, v této funkci jsou registrovány dostupné url pomocí `httpd_register_uri_handler` a jednotlivých cest: `/` pro hlavní stránku - `root_handler()`, `/historie.txt` pro textový soubor s logy teplot - `get_file_handler()`, `/submit` pro odeslání formuláře a přijetí dat - `submit_handler()`, a `/` pro zobrazení 404 - výchozí `httpd_resp_send_404`. Všechny metody jsou GET. Ve funkci `submit_handler()` se z URL vyparsují znaky s informacemi a podle klíčů `value` a `hist` se vybere hodnota pro teplotu a histerzi. Do konzole se vypíše informace o tom, že byla daná hodnota uložena, pokud je histerze menší než 0, hodnota se nastaví na 0. Funkce `get_file_handler()` otevře soubor `historie.txt` pro čtení a postupně pomocí `httpd_resp_send_chunk()` zašle část dat ze souboru z bufferu o velikosti 1024, poté soubor uzavírá. Hlavní stránku zobrazuje `root_handler`, ten nahradí proměnné v připravené HTML šabloně reálnými daty a zašle klientovi. Nazrazují se informace o aktuální teplotě, hodnoty prahové teploty a histerze.

### 3.5 Měření teploty

Další v pořadí se nastaví nakonfiguruje ADC převodník<sup>3</sup> pro získávání hodnot z teplotního čidla. Zde jsem narazil při kompilování na varování, že se jedná o zastaralou knihovnu a je vhodné ji nahradit knihovnou novější, ta mi však bohužel nefungovala a z toho důvodu jsem se rozhodl zůstat u této starší verze. Pro inicializaci jsem využil funkce `esp_adc_cal_characterize()`, která zapíše kalibrační data do struktury `adc1_chars` pro přesnější data. Pro převod hodnot z čidla jsem využil funkce `esp_adc_cal_raw_to_voltage()` která tato data využívá pro přesnější získávání hodnot v mV. Výsledná teplota je potom vypočtena pomocí vztahu z dokumentace čidla [2][Strana 10].

### 3.6 Synchronizace s NTP serverem

Pro zápis do historie je třeba časová značka z doby pořízení zápisu, pro tento účel inicializuji čas ESP na výchozí hodnotu, kterou se následně pokusím aktualizovat pomocí volání na server `tik.cesnet.cz`, k tomu soužij nastavení serveru v `esp_ntp_config_t` a zavolání funkce `esp_netif_ntp_init()` s touto strukturou a následně se pokusím kontaktovat server `esp_netif_ntp_sync_wait()` s časováním na

<sup>2</sup><https://www.freertos.org/a00125.html>

<sup>3</sup><https://docs.espressif.com/projects/esp-idf/en/v4.4/esp32/api-reference/peripherals/adc.html#adc-api-reference-adc-driver>



max 15 vteřin, po té době se vypíše hláška o neúspěchu a pokračuje se v běhu programu. V případě úspěchu se opět vypíše hláška o úspěšné aktualizaci času.

### 3.7 LED Dioda

LED dioda je zapojená viz 2.2. Její ovládání spočívá v porovnání nastavené prahové teploty a histerze s reálnou hodnotou, pokud je aktuální teplota nižší nebo rovna teplotě (nastavená - histerze), pak se dioda sepne, pokud je teplota vyšší než (nastavená + histerze) naopak se vypne. V případě, že je teplota mezi těmito hodnotami, s diodou se nic neděje a zůstává ve stejném stavu.

### 3.8 Hlavní smyčka

Po nastavení ADC převodníku se vstupuje do hlavní smyčky programu `while(1)`, kde se každý cyklus kontroluje naměřená hodnota z čidla, vypočítává se teplota, získává aktuální čas a vše se vypisuje na výstup, zároveň také zapisuje do souboru `historie.txt`. Poté smyčka vyčkává pomocí `vTaskDelay` 1 vteřinu a pokračuje v dalším cyklu.

## 4 Testování

Pro ověření hodnot jsem využíval multimetr a testování probíhalo pomocí vypisování debug výstupů do konzole.

## 5 Závěr

Podařilo se mi implementovat všechny části aplikace, které byly uvedené v zadání projektu. Při testování projektu jsem narazil pouze na nižší výstupní hodnotu z čidla, která byla následně zkalibrována na správný výsledek pomocí ESP. Celkově jsem nenarazil na žádné chybné chování programu, v pořádku proběhlo nastavení, nahrání, spuštění všech částí projektu i ovládání pomocí webserveru a ukládání historie dat. Čas zařízení je synchronizován ze serveru `tik.cesnet.cz`. Jako literaturu jsem použil zadanou studijní [2] a dokumentaci pro ESP IDF [1].

## 6 Video

Pro předvedení fungování mého projektu jsem natočil 2 videa, krátké dle zadání, a dlouhé, ve kterém vysvětluji podrobněji fungování projektu. Budu rád za zhlédnutí. <https://nextcloud.fit.vutbr.cz/s/oTT3ZyjFBEWSytn>

## Reference

- [1] Espressif: *API Reference*. Dostupné z: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference>
- [2] Instruments, T.: *LMT85 1.8-V, SC70/TO-92/TO-92S, Analog Temperature Sensors*. Dostupné z: <https://www.ti.com/lit/ds/symlink/lmt85.pdf>