



IMP projekt 2023

Meteostanice

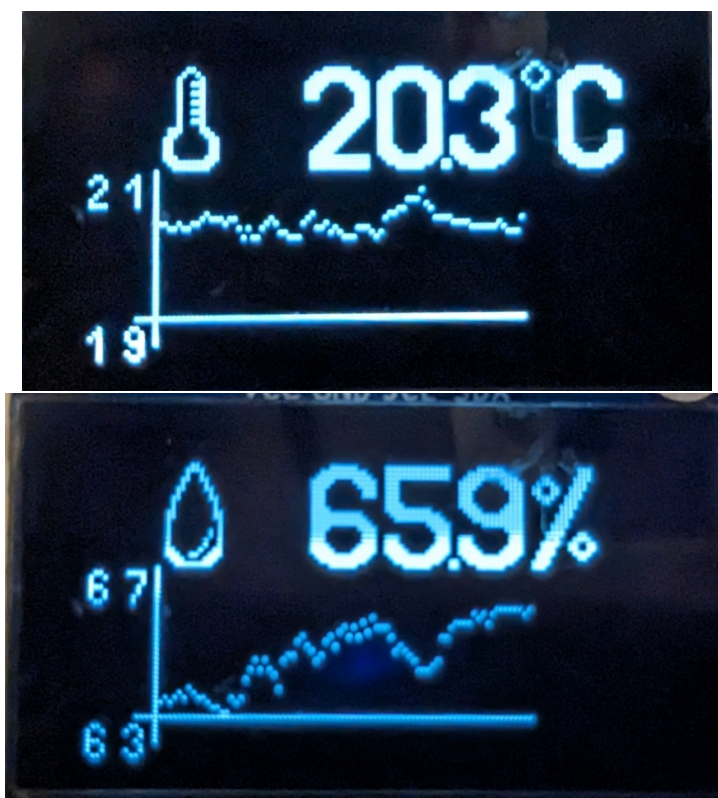
Obsah

1	Úvod	2
2	Schéma zapojení	3
3	Konfigurace	3
4	Senzor SHT31	3
5	Displej SSD1306	4
6	Implementace	4
6.1	Inicializace I ² C rozhraní	4
6.2	Měření teploty a vlhkosti	4
6.3	Ukládání naměřených hodnot do grafu	5
6.4	Zobrazování naměřených dat v grafu	5
6.5	Zobrazení dat na displej	5

1 Úvod

Cílem projektu je vypracovat měření teploty a vlhkosti senzorem SHT31 a zobrazovat naměřené informace na displeji SSD1306, veškerá logika je řízena mikrokontrolerem ESP32. K vypracování řešení bylo zvoleno prostředí idf¹.

Výsledné řešení po spuštění volitelně přehraje animaci a začne periodicky zobrazovat údaje o teplotě a vlhkosti, společně s grafem naměřených hodnot. Graf i zobrazená hodnota se aktualizují vždy před zobrazením příslušných údajů. Frekvence přepínání obrazovek je konfigurovatelná před sestavením aplikace a ovlivňuje tedy i frekvenci zaznamenávání dat do grafu.

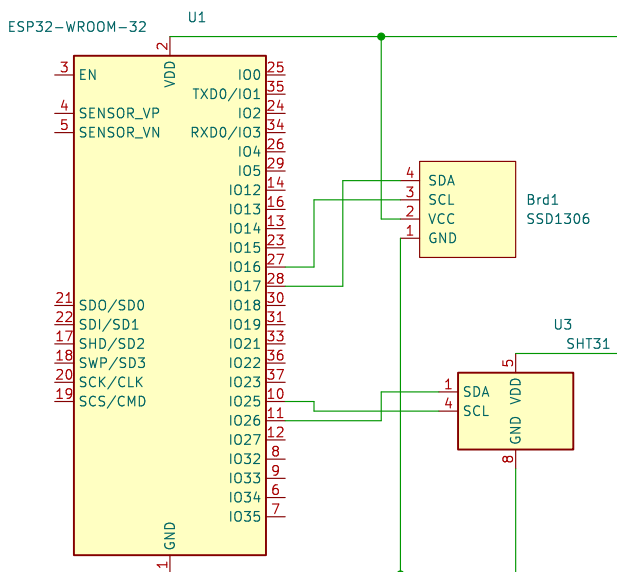


Obrázek 1: Ukázka naměřených hodnot s intervalem měření 2 minuty

¹<https://github.com/espressif/esp-idf>

2 Schéma zapojení

Schéma obsahuje výchozí čísla pinů, která lze změnit, viz sekce 3.



Obrázek 2: Schéma zapojení
pozn. vcc v tomto případě odpovídá 3v3 vcc

3 Konfigurace

Konfigurace programu probíhá před sestavením přes vestavěné rozhraní `menuconfig`. Lze povolit nebo zakázat úvodní animaci, nastavit periodu přepínání obrazovek s teplotou a vlhkostí a nastavit konfiguraci jednotlivých periférií, konkrétně čísla GPIO pinů a pracovní frekvenci. Všechna nastavení jsou dostupná z hlavní strany pod `IMP project configuration`.

4 Senzor SHT31²

Senzor SHT31 pro měření teploty a vlhkosti podporuje režim jednorázového a periodického měření, pro projekt byl využit pouze jednorázový režim. Měření v tomto případě probíhá tak, že mcu zašle 2-bytový příkaz, který specifikuje, že se jedná o jednorázové měření a parametr opakovatelnosti měření (viz dokumentace k senzoru), počká požadovanou dobu (dle zvolené opakovatelnosti) a přečte data z I²C rozhraní. Formát odpovědi je 6 bytů, postupně 2 byty hodnoty teploty, kontrolní byte pro teplotu, 2 byty hodnoty vlhkosti a kontrolní byte pro vlhkost. Obě hodnoty jsou ve floating point formátu.

Získané hodnoty je potřeba zkontrolovat pomocí `crc`³ a přepočíst na finální hodnotu teploty a vlhkosti podle vzorce z dokumentace.

²https://www.laskakit.cz/user/related_files/sht3x.pdf

³https://en.wikipedia.org/wiki/Cyclic_redundancy_check

5 Displej SSD1306⁴

Pro řešení projektu je použito pouze rozhraní pro zapisování dat přes I²C do displeje, avšak displej podporuje i mód čtení, především pro zjištění stavu paměti, který ale nebyl potřeba.

Zapisování dat probíhá ve 2 režimech – pro zasílání příkazů a dat, z nichž každý má variantu pro jeden byte nebo nepřetržitý tok dat. Řešení projektu používá pouze variantu pro tok dat (stream). Režim je určen prvním bytem (nepočítáme byte s adresou a režimem I²C) a zbytek už je určen pro data nebo příkazy (toto platí pouze pro stream varianty).

Příkazy (commands) slouží především ke konfiguraci a ovládání displeje. V projektu jsou použity pro inicializaci displeje, avšak je možné posílat příkazy za běhu a tím měnit např. režimy zobrazení, adresu, na kterou se zapisují data apod. Inicializace využívá sekvenci příkazů popsanou v dokumentaci (Application note) s několika úpravami. Nejdůležitější z nich je nastavení horizontálního adresovacího režimu paměti, který zajistí správné namapování bufferu s daty do RAM paměti displeje díky automatické inkrementaci čítače zapisované adresy, která se zalamuje na konci řádků displeje.

Data určují, co bude zobrazeno na displeji. Displej má 128x64 pixelů a je rozdělen na strany – strana obsahuje 8 pixelů na výšku, displej má tedy dohromady 8 stran a každá strana obsahuje 128 sloupců po 8 pixelech. Data jsou uložena ve struktuře `display_buff`, která odpovídá osmi polím/stranám po 128 bytech.

6 Implementace

Implementace je v jazyce C pouze s využitím základních knihoven a driveru k I²C rozhraní⁵. Veškeré ovládání periférií je implementováno pouze s využitím tohoto rozhraní. V následujících sekcích je postupně popsány jednotlivé kroky programu.

6.1 Inicializace I²C rozhraní

Pro práci se zařízeními je použita struktura `dev_conf_t`, která obsahuje I²C adresu a port, frekvenci hodinového signálu a čísla pinů pro hodinový signál a datový tok. Inicializace probíhá ve funkci `i2c_init()`, která nastaví potřebné hodnoty a zavolá funkci pro instalaci I²C driveru.

6.2 Měření teploty a vlhkosti

Hlavní smyčka programu volá funkci pro měření a zobrazení teploty, poté počká zvolenou dobu, zavolá funkci pro měření a zobrazení vlhkosti, opět počká a celá smyčka se opakuje.

Zobrazení veškerých znaků je řešeno zápisem dat z hlavičkového souboru `glyphs.h` do bufferu, který je nejprve naplněn daty a poté poslán přes I²C rozhraní do displeje jako data (viz sekce 5). Slouží k tomu především funkce `write_to_buff()`, která na danou stranu a x-ovou souřadnici zapíše data o dané šířce a výšce. Šířka je v jednotkách, které odpovídají jednomu sloupci dat/pixelů na straně displeje, výška udává počet stran, které se mají zapsat. Jednotlivé glyfy jsou uloženy jako surová data, která odpovídají jejich zobrazení na displeji, tedy LSB odpovídá prvnímu pixelu na dané straně a MSB tomu poslednímu. Všechny glyfy jsou vlastnoručně vytvořené pomocí přiložené tabulky `glyph-maker.ods`.

⁴https://www.laskakit.cz/user/related_files/ssd1306.pdf

⁵<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/i2c.html>

Měření probíhá ve funkci `sht31_get_data()`, která změří data výše popsaným způsobem a vrací strukturu s naměřenou teplotou a vlhkostí. Data se přidávají do dat grafu, případně se aktualizuje maximální a minimální hodnota naměřených dat.

6.3 Ukládání naměřených hodnot do grafu

Struktura dat grafu si ukládá současný index, minimální a maximální hodnotu a informaci o tom, jestli došlo k naplnění maximálního počtu dat grafu, který je specifikován v souboru `utils.h` pod `MAX_GRAPH_DATA`.

Dokud není struktura plná, data se dále přidávají dokud nedojde k naplnění struktury a přitom se aktualizuje maximum a minimum, v tomto případě stačí porovnat novou hodnotu se stávajícím minimem a maximem. Když dojde k naplnění struktury, nastaví se příznak naplnění a při dalším přidání hodnoty do grafu je potřeba naměřená data posunout (zahodí se nejstarší hodnota a přidá se nová). Pro vypočítání nového maxima a minima je nutné projít všechna data, protože minimum a maximum už nemusí být aktuální.

6.4 Zobrazování naměřených dat v grafu

Zápis dat grafu do bufferu provádí funkce `draw_graph()`. Základní je zjištění rozsahu hodnot dat pro zobrazení, které odpovídá $\lceil v_{max} \rceil - \lfloor v_{min} \rfloor$, pokud předpokládáme, že v jsou naměřené hodnoty. Pokud by měl být rozsah příliš malý (stanoveno hodnotou `GRAPH_DATA_MIN_RANGE`) nastaví se na minimální rozsah. Podle zvoleného rozsahu dat se dopočítají hranice rozsahu zaokrouhlené na celá čísla.

Zvolené hranice se zapíše do bufferu vedle os grafu a začnou se mapovat a zapisovat jednotlivá data. Data se mapují tak, aby nejnižší bod grafu odpovídal zvolené spodní hranici a nejvyšší bod grafu zase zvolené horní hranici. Pro každou hodnotu je nutné vypočítat vzdálenost v pixelech od spodní hranice, v tomto případě dochází k přepočítávání x a y souřadnic na strany a nastavuje se konkrétní pixel ve sloupci strany. Mapování probíhá podle následujícího vzorce

$$(v_i - lower_bound) * \frac{GRAPH_POINT_MAX_RANGE}{data_range}$$

6.5 Zobrazení dat na displej

V tento moment už jsou v bufferu zapsána všechna data pro vykreslení a celý buffer se jako data pošle přes I²C rozhraní displeji, viz sekce 5.