

KATOLÍCKA UNIVERZITA V RUŽOMBERKU
NÁZOV FAKULTY

SmartAir-monitor pomocou ESP32S

Projektová dokumentácia

2025

Ing., Mgr. Rita Kost'uvá

Cieľ:

Cieľom mojej semestrálnej práce bolo navrhnuť a vytvoriť **SmartAir monitor**, zariadenie na monitorovanie kvality ovzdušia, ktoré dokáže poskytovať presné údaje o koncentrácii znečisťujúcich látok v reálnom čase. Tento projekt bol zvolený s ohľadom na rastúcu potrebu sledovania znečistenia ovzdušia v mestských aj domácich prostrediach, kde kvalita vzduchu priamo ovplyvňuje zdravie a komfort obyvateľov.

Pri návrhu a realizácii som spojila viacero senzorov a komponentov. **SmartAir monitor** sleduje najdôležitejšie ukazovatele znečistenia ovzdušia:

- CO₂ (oxid uhličitý)
- NH₃(amoniak)
- NO_x(oxid dusíka)
- SnO₂(oxid cínatý)
- CO (oxid uhoľnatý)
- NO₂ (oxid dusičitý)
- Jemné prachové častice (PM_{2.5})

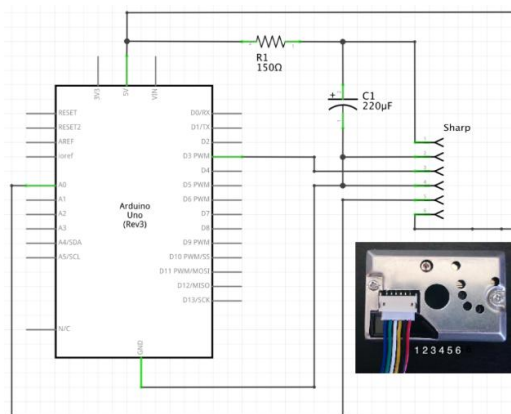
Údaje sú zobrazované na OLED displeji, aby boli ľahko prístupné pre používateľa. Každý parameter je prehľadne zobrazený spolu s jeho aktuálnou hodnotou v reálnom čase, čo používateľovi umožňuje okamžite sledovať kvalitu vzduchu. Navyše, celý systém je postavený na mikrokontroléri ESP32, ktorý po 20 sekundách sa resetuje formou blikania na displayi.

Súčiastky:

1. Breadboard – prepojovacie pole
2. ESP-32S 30P Expansion board
3. OLED 0.96" display 128x64 - 4 piny
4. MQ7
5. MQ135
6. GP2Y1014AU0F Compact Optical Dust Sensor
7. Prepájacie drôty
8. rezistor
- 9.kondenzátor
10. Kábel USB-C

Postup práce:

1. Z ESP 32 som vyviedla napätie 5V na
 - MQ7 na detekciu toxických koncentrácií
2. Z ESP 32 som vyviedla napätie 3V na merač
 - MQ135-Oxid uhličitý a oxid dusičitý
 - OLED 0.96" display
3. Z ESP 32 GND káblíky čiže čierne žena-žena som prepojila s MQ135+MQ7+Oled displayom
4. Na fotkách, ktoré sú vo fotodokumentácii je vidieť množstvo káblov, kde červený je použitý pre napájanie a čierny je použitý pre GND čiže uzemnenie
5. MQ135 som napojila žltým káblíkom žena-žena z AO na GPIO35
6. MQ 7 som takisto napojila žltým káblíkom žena-žena z AO na GPIO 34
7. Ako posledné som pripojila USB-C kábel do ESP 32 s PC čo mi aj rozsvietilo červené svetielko
8. Ako ďalší postup som do počítača stiahla Arduino IDE 2.3.4 a stiahla si ešte do počítača na ESP 32 Silicon Labs Downloads CP 210x drivers
9. Po úspešnom trápení a zapojení som prešla k písaniu kódu
10. Najprv som otestovala Display lebo sa mi tam nič nezobrazovala a ani sa nezaplo ale správnym kódom nakoniec vypísalo (Hello Oled)
11. Po úspešnom zapojení som prešla k písaniu kódu, ktorý je viditeľný v nasledujúcej časti - NAPÍSANIE KÓDU
12. Keďže som mala čas ešte na tretiu vec a to na merač prachu. Bola som toho názoru, že budem šťastná ak vôbec mi pôjdu tie dve senzory MQ135 a MQ7 a keď mi to bude fungovať a zostane čas tak prejdem na merač prachu. Z ESP 32 som vyviedla napätie 5V červený káblík na merač prachu GP2Y1014AU0F Compact Optical Dust Sensor a GND na uzemnenie čierny káblík
13. Z GP2Y1014AU0F(merač prachu) som postupovala podľa nasledujúceho obrázka



Zapojíme 1 cez odpor na +5V, 2 na zem, 3 na digitálny pin 3, 4 na zem, 5 na analógový pin 0 a posledné 6 na +5V. Pripojila som k tomu kondenzátor podľa vyššej schémy, čiže kladný pól na 1 a záporný pól na zem.

14. Z GP2Y1014AU0F(merač prachu) som VCC červený káblík žena-žena prepojila na ESP32 na 5V, čierny káblík žena-žena na GND do ESP32.
15. GP2Y1014AU0F(merač prachu) oranžový káblík s označením A ako analógový som na ESP32 na GPIO D26. Biely káblík s označením D ako digitálny na GPIO D5.
16. Následne podľa kódu nižšie som spojila v Arduino

Napísanie kódu:

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

// Definovanie rozmerov displeja
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

// Vytvorenie objektu pre displej
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

// Definovanie pinov pre senzory
#define MQ135_PIN 35 // Pin pre MQ135 (CO2, NH3, NOx, SnO2)
#define MQ7_PIN 34 // Pin pre MQ7 (CO)
#define DUST_LED_PIN 5 // Pin pre LED indikátor GP2Y1014AU0F
#define DUST_ANALOG_PIN 26 // Pin pre analógový výstup GP2Y1014AU0F

// Funkcia na prevod analógového čítania na koncentráciu CO2 v ppm (MQ135)
float convertToPPM_MQ135(int analogValue) {
  float voltage = (analogValue / 4095.0) * 3.3; // Prevod na napätie
  float ppm = voltage * 100; // Tento výpočet je orientačný, kalibrácia je potrebná
  return ppm;
}

// Funkcia na prevod analógového čítania na koncentráciu NH3 v ppm (MQ135)
float convertToPPM_NH3(int analogValue) {
  float voltage = (analogValue / 4095.0) * 3.3; // Prevod na napätie
  float ppm = voltage * 50; // Kalibrujte podľa datasheetu
  return ppm;
}

// Funkcia na prevod analógového čítania na koncentráciu NOx v ppm (MQ135)
float convertToPPM_NOx(int analogValue) {
  float voltage = (analogValue / 4095.0) * 3.3; // Prevod na napätie
  float ppm = voltage * 80; // Kalibrujte podľa datasheetu
  return ppm;
}

// Funkcia na prevod analógového čítania na koncentráciu SnO2 v ppm (MQ135)
float convertToPPM_SnO2(int analogValue) {
  float voltage = (analogValue / 4095.0) * 3.3; // Prevod na napätie
  float ppm = voltage * 40; // Kalibrujte podľa datasheetu
  return ppm;
}

// Funkcia na prevod analógového čítania na koncentráciu CO v ppm (MQ7)
float convertToPPM_MQ7(int analogValue) {
  float voltage = (analogValue / 4095.0) * 3.3; // Prevod na napätie
  float ppm = voltage * 50; // Tento výpočet je orientačný, kalibrácia je potrebná
  return ppm;
}

// Funkcia na prevod analógového čítania na hodnotu prachu v µg/m3 (GP2Y1014AU0F)
float convertToDust(int analogValue) {
  float voltage = (analogValue / 4095.0) * 3.3; // Prevod na napätie
  float dust = voltage * 500; // Tento výpočet je orientačný, kalibrácia je potrebná
  return dust;
}

unsigned long previousMillis = 0; // Časovač pre blikanie displeja
const unsigned long interval = 2000; // 20 sekúnd

void setup() {
  // Inicializácia sériovej komunikácie
```

```

Serial.begin(115200);

// Inicializácia displeja
if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("OLED initialization failed!");
    while (true);
}

// Inicializácia pinov
pinMode(DUST_LED_PIN, OUTPUT);
digitalWrite(DUST_LED_PIN, HIGH); // Zapnutie LED pre GP2Y1014AU0F

// Vymaž displej
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
}

void loop() {
    // Čítanie hodnoty z MQ135
    int mq135_value = analogRead(MQ135_PIN);
    float co2_ppm = convertToPPM_MQ135(mq135_value);
    float nh3_ppm = convertToPPM_NH3(mq135_value);
    float nox_ppm = convertToPPM_NOx(mq135_value);
    float sno2_ppm = convertToPPM_SnO2(mq135_value);

    // Čítanie hodnoty z MQ7 (CO)
    int mq7_value = analogRead(MQ7_PIN);
    float co_ppm = convertToPPM_MQ7(mq7_value);

    // Čítanie hodnoty prachu z GP2Y1014AU0F
    int dust_value = analogRead(DUST_ANALOG_PIN);
    float dust_ug_m3 = convertToDust(dust_value);

    // Časovač pre blikanie displeja
    unsigned long currentMillis = millis();
    if (currentMillis - previousMillis >= interval) {
        previousMillis = currentMillis;

        // Displej sa vypne na 1 sekundu
        display.clearDisplay();
        display.display();
        delay(1000);

        // Displej zobrazí bielu obrazovku na 3 sekundy
        display.clearDisplay();
        display.fillScreen(SSD1306_WHITE);
        display.display();
        delay(3000);
    }

    // Zobrazenie hodnôt na displeji
    display.clearDisplay();
    display.setCursor(0, 0);
    display.print("CO2: ");
    display.print(co2_ppm);
    display.println(" ppm");

    display.setCursor(0, 10);
    display.print("NH3: ");
    display.print(nh3_ppm);
    display.println(" ppm");

    display.setCursor(0, 20);
    display.print("NOx: ");
    display.print(nox_ppm);
    display.println(" ppm");
}

```

```
display.setCursor(0, 30);
display.print("SnO2: ");
display.print(sno2_ppm);
display.println(" ppm");

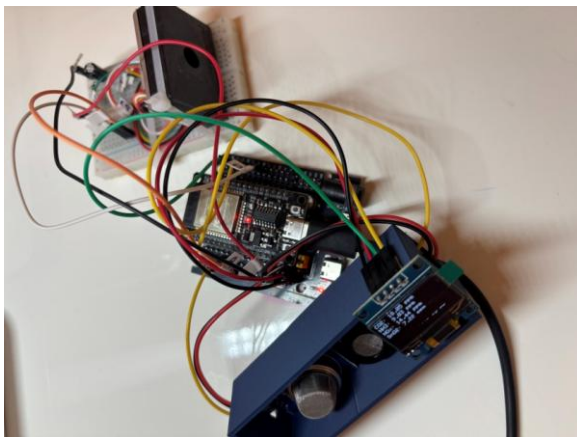
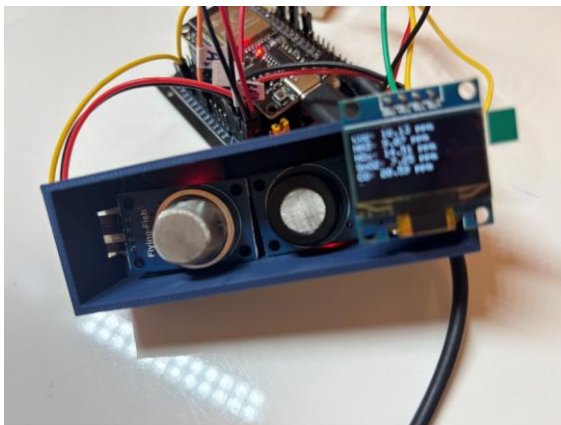
display.setCursor(0, 40);
display.print("CO: ");
display.print(co_ppm);
display.println(" ppm");

display.setCursor(0, 50);
display.print("Dust: ");
display.print(dust_ug_m3);
display.println(" ug/m3");

// Aktualizácia displeja
display.display();

// Pauza medzi čítaniami
delay(1000); // Čítať každú sekundu
}
```

Fotodokumentácia:



Link na GitHub:

Zdroj:

<https://www.hackster.io/512389/esp32-monitoring-air-quality-with-mq-135-integrated-blynk-77d18b>

<https://blog.asksensors.com/air-quality-sensor-mq135-cloud-mqtt/>

<https://navody.drtek.cz/navody-k-produktum/opticky-senzor-kvality-ovzdusi-sharp-gp2y1010au0f.html>

<https://blog.asksensors.com/air-quality-sensor-mq135-cloud-mqtt/>

<https://navody.drtek.cz/navody-k-produktum/senzor-plynu-mq-135.html>