

Programowanie sieciowe

Grupa Numer: 7 Czw. 18:30 Jakub Łudzik Paweł Michalcewicz Filip Żurek	Projekt Usługa do zdalnego pobierania odczytu czujnika BME680 na ESP32	Data przesłania projektu: 25.01.2025
----------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------	----------------------------------------------------

1. Struktura projektu

Serwer: ESP32, współbieżny, statyczne IPv4, nasłuchuje na porcie 9999

Pliki:

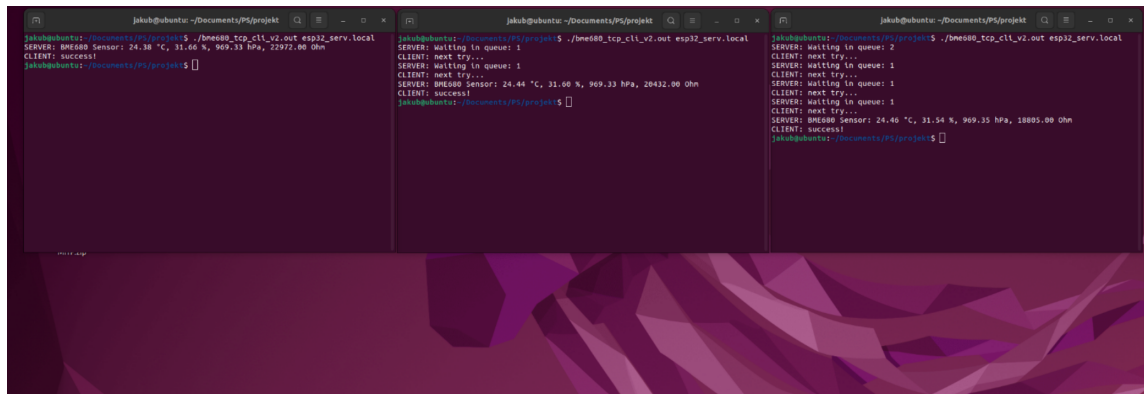
- `main.c` – główny kod serwera
- `myunp_esp32.h` – funkcje „opakowujące” na podstawie R.Stevens *Unix Network Programming*
- katalog `components` – biblioteka do obsługi czujnika BME680

Klient: Ubuntu, łączenie z serwerem przez dns: `esp32_serv.local` lub IPv4

Pliki:

- `bme680_tcp_cli.c` – główny kod klienta
- `myunp.h` – funkcje „opakowujące” na podstawie R.Stevens *Unix Network Programming*

2. Prezentacja działania



Rysunek 1: Łączenie z serwerem za pomocą 3 klientów

3. Opis kodu

Ze względu na brak możliwości tworzenia procesów potomnych na ESP32 nie było możliwe zrealizowanie współbieżności za pomocą funkcji `fork()`. W tym celu wykorzystano dynamiczną alokację pamięci dla deskryptora gniazda połączonego.

Nie jest możliwy jednoczesny odczyt danych z czujnika jednak, aby projekt był zgodny z wymaganiami dotyczącymi współbieżności możliwe jest jednoczesne połączenie kilku klientów. W takiej sytuacji utworzona zostaje kolejka i odczyty są przesyłane do klientów zgodnie z kolejnością łączenia z serwerem, a klient pozostając w sesji TCP ponawia próbę odczytu z czujnika co sekundę.

Serwer:

```
void mainServerTask(void* arg)
```

- utworzenie zmiennej dla deskryptora gniazda nasłuchującego
- utworzenie zmiennej dla deskryptora gniazda połączonego jako wskaźnik na `int`
- utworzenie i wypełnienie struktury `sockaddr_in`
- powiązanie struktury z deskryptorem gniazda nasłuchującego
- rozpoczęcie nasłuchiwanie na porcie 9999

- w pętli akceptującej połączenia przychodzące dynamiczna alokacja pamięci dla tego deskryptora za pomocą funkcji `malloc()`
- funkcja `Accept()` zwraca deskryptor do gniazda zapisując go pod poprzednio zarezerwowanym adresem
- utworzenie `clientTask` przekazując mu jako argument wskaźnik do deskryptora gniazda

`void clientTask(void* arg)`

- przekazanie deskryptora do nowej zmiennej
- zwolnienie pamięci w celu przyjęcia następnych klientów
- zebranie pomiarów i odesłanie do klienta, który jest pierwszy w kolejce

Klient:

- połączenie z serwerem za pomocą funkcji `Connect()`
- odczyt w pętli za pomocą funkcji `Readn()`

4. Rozkład pracy w grupie

- Jakub Łudzik – serwer, ESP32
- Filip Żurek – klient, DNS
- Paweł Michalcewicz – obsługa czujnika, obsługa błędów funkcji sieciowych (funkcje „opakowujące”)