

0.1 Założenia projektowe

- Funkcjonalność ustawiania godziny budzika będzie realizowana przez zewnętrzny serwer.
- Na wyświetlaczu Nixie będą wyświetlane godziny, minuty, sekundy.
- Od spodu obudowy będą umieszczone paski LED, które będą podświetlały obudowę i będą wyświetlane animacje podczas alarmu.
- Alarm będzie sygnalizowany dźwiękiem oraz miganiem pasków LED.
- Wyłączanie alarmu będzie możliwe poprzez przycisk na obudowie, aplikację mobilną lub zewnętrzny przycisk połączony z serwerem.
- W przypadku braku połączenia z serwerem, czas będzie mierzony przez RTC wbudowany w mikrokontroler.
- Programowa oraz manualna regulacja jasności wyświetlacza Nixie oraz pasków LED.

```
graph TD; Zasilanie[Zasilanie] --> Złącze_zasilania[Złącze zasilania]; Zasilanie --> Lampy_nixie[Lampy nixie]; Zasilanie --> Mikrokontroler[Mikrokontroler]; Mikrokontroler --> Złącze_programownia[Złącze do programownia]; Mikrokontroler --> Złącze_debugowania[Złącze do debugowania]; Mikrokontroler --> Pasek_LED[Pasek LED]; Mikrokontroler --> Glosnik[Głośnik/Buzzer]; Mikrokontroler --> Sterowanie[sterowanie lampami]; Sterowanie --> Lampy_nixie; Interfejs_uzytkownika[Interfejs użytkownika] --> Mikrokontroler;
```

Sekcja opisana jako *zasilanie* będzie odpowiedzialna za zasilanie wszystkich elementów układu, w tym lamp Nixie, paska LED, mikrokontrolera oraz głośnika, więc będzie wymagane rozbić jej

na kilka podsekcji, ponieważ będą potrzebne różne napięcia zasilania. Lampy Nixie potrzebują zasilania wysokim napięciem, natomiast pozostałe elementy potrzebują zdecydowanie niższych napięć. Blok *sterowanie lampami Nixie* będzie odpowiedzialny za wyświetlanie odpowiednich cyfr na lampach. Sekcja *Interfejs użytkownika* będzie odpowiedzialna za interakcję z użytkownikiem, w tym regulacja jasności oraz wyłączania alarmu, ważne by interfejs był intuicyjny i jak najbardziej rozwojowy na potencjalne przyszłe funkcje.

0.2 Ogólny schemat komunikacji z serwerem

Urządzenie będzie musiało komunikować się z serwerem czasu, a także z serwerem Home Assistant, który będzie interfejsem użytkownika. Komunikacja z serwerem czasu będzie odbywała się poprzez protokół NTP, ponieważ inne protokoły opisane w rozdziale ?? służą do zapewnienia większej dokładności czasu, co nie jest wymagane w tym projekcie. Inne protokoły wymagają też większej ilości zasobów lub specjalistycznego sprzętu. Kolejną zaletą wyboru NTP jest to, że jest to najbardziej popularny protokół do synchronizacji czasu w sieciach komputerowych, co sprawia, że jest on najbardziej przetestowany i stabilny. Posiada on wiele implementacji, które są dostępne na wielu platformach, w tym na platformę ESP32. Komunikacja z serwerem Home Assistant będzie odbywała się poprzez protokół MQTT, który jest bardzo popularnym protokołem w IoT, co pozwoli na łatwe rozbudowanie funkcjonalności. Połączenia te przedstawione są na rysunku 0.2.

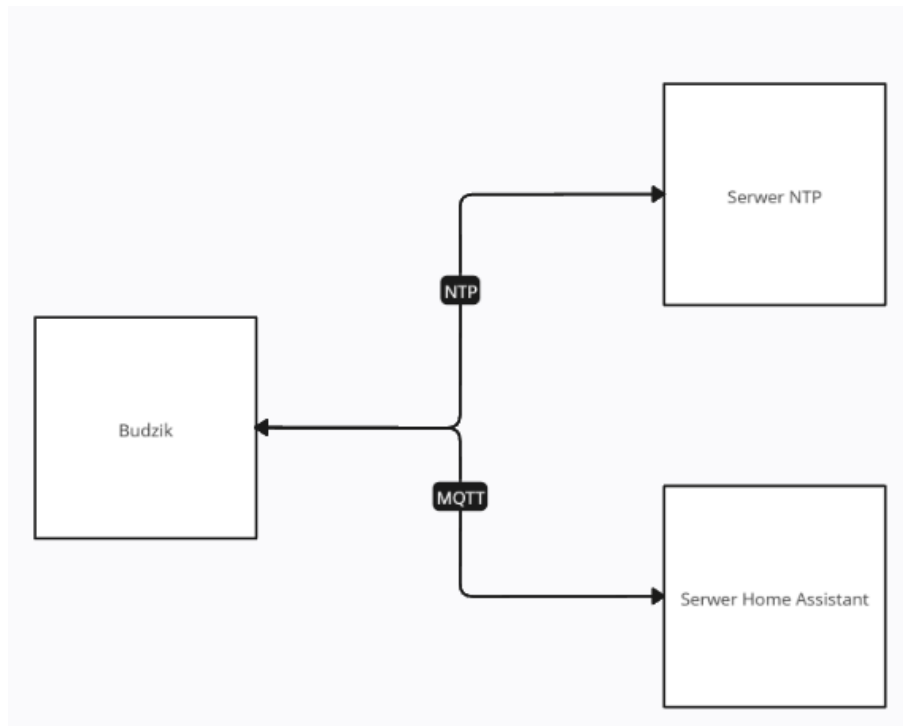


Figure 0.2: Ogólny schemat komunikacji z serwerem