**Dokumentacja wstępna:**

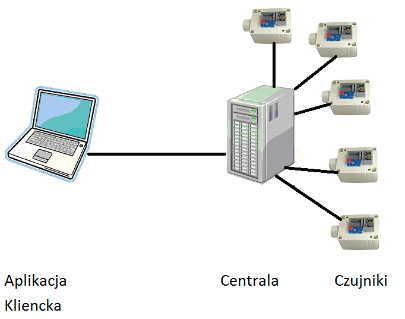
**Skład zespołu:**

* Jakub Dudziak
* Paweł Joński
* Michał Sacharczuk - Lider Zespołu
* Michał Tomaszewski

**Prowadzący:** dr inż. Jacek Wytrębowicz

Projekt wstępny powinien określać:

1. **Treść zadania.**
   1. Proszę założyć istnienie sieci specjalizowanych czujników do wykrywających zmiany rezystancji między drutem oporowym a kontrolowanym fragmentem rury. Czujniki taki może zgłosić alarm do centrali i podać wartości rezystancji typowej dla danego odcinka i bieżącej. Próg dla zgłaszania alarmu jest programowany przez centralę. W centrali znajduje się serwer nadzorujący pracą czujników. Operator współpracuje z serwerem poprzez: 1) skrypty powłoki, 2) przeglądarkę WWW. Rozważyć funkcjonowanie centrali zapasowej. Należy zaprojektować i wykonać system komunikacji pomiędzy elementami wymienionej tu infrastruktury. Należy przewidzieć modyfikacje tej infrastruktury w czasie jej eksploatacji. System komunikacji powinien móc pracować w przestrzeni adresów IPv4 i IPv6. Ponadto należy zaprojektować moduł do Wireshark umożliwiający wyświetlanie i analizę zdefiniowanych komunikatów.
2. **Nazwę własną projektowanego systemu.**
   1. eRURA
3. **Przyjęte założenia funkcjonalne i niefunkcjonalne**.
   1. Założenia funkcjonalne
      1. System
         1. System komunikacji powinien móc pracować w przestrzeni adresów IPv4 i IPv6.
         2. System umożliwia dodawanie czujników w trakcie jego działania
         3. W ramach systemu będzie funkcjonowała centrala zapasowa
      2. Centrala
         1. Zadaje poszczególnym czujnikom rezystancje alarmowe.
         2. Na żądanie operatora sprawdza rezystancję na wybranym odcinku.
         3. Przechowuje historie alarmów.
      3. Czujnik:
         1. Sprawdza rezystancję na swoim odcinku rury i porównuje ją z rezystancją alarmu
         2. Zgłasza alarm kiedy rezystancja spadnie poniżej rezystancji alarmowej (przeciek) lub gdy wzrośnie do nieskończoności (zerwany drut).
      4. Przeglądarka:
         1. Umożliwia wyświetlenie listy czujników.
         2. Umożliwia zarządzanie poziomem alarmów.
         3. Wyświetla alarmy.
         4. Daje możliwość przeglądania historii alarmów.
         5. Wyświetla statystyki.
      5. Skrypt powłoki
         1. Umożliwia wyświetlenie listy czujników.
         2. Umożliwia zarządzanie poziomem alarmów.
         3. Daje możliwość przeglądania historii alarmów
         4. Pozwala na usunięcie historii alarmów.
         5. Wyświetla alarmy.
   2. Założenia niefunkcjonalne:
      1. Umożliwia obsługe znacznej ilości czujników.
      2. Intyuicyjna obsługa przez przeglądarkę.
      3. Bezpieczeństwo użytkowania.
      4. Centala oparta na wątkach.
4. **Podstawowe przypadki użycia.**
   1. PU1. Wysłanie informacji o nowym stanie alarmu.
   2. PU2. Wyświetlenie listy czujników.
   3. PU3. Wyświetlenie stanu instalacji.
   4. PU4. Przeglądanie historii alarmów
   5. PU5. Usunięcie historii alarmów.
5. **Wybrane środowisko sprzętowo-programowe (systemy operacyjne, biblioteki programistyczne) i narzędziowe (debugowanie, testowanie).**
   1. System operacyjny - linux
   2. Biblioteki programistyczne
      1. Gniazda BSD sys/socket.h
   3. IDE CLion
   4. Build system - CMake
   5. Debugowanie - GDB
   6. Testowanie Cppunit lub Google Test
6. **Architekturę rozwiązania, tj. ilustrację i opis struktury logicznej systemu (koncepcyjnych bloków funkcjonalnych).**
   1. Czujnik jako serwer.
   2. Centrala jako klient.
   3. Obsługa modułu http



1. **Sposób testowania.**
   1. Testowanie jednostkowe nowych funkcjonalności.
   2. Testowanie poszczególnych modułów.
   3. Testowanie integracyjne.
   4. Testowanie funkcjonalne.
2. **Sposób demonstracji rezultatów, tj. scenariusze testów akceptacyjnych do zaprezentowania przy odbiorze projektu.**
   1. Scenariusz 1:
      1. Skrypt uruchamia instalacje z zadaną ilością czujników.
      2. Dla wszystkich ustawiania jest wartość alarmowa 5.
      3. Za pomocą skryptu na kolejnych czujnikach symulaowany jest przeciek poprzez obniżenie wartości rezystancji poniżej wartości alarmowej o 1.
      4. Oczekiwany rezultat - alarm na wszystkich czujnikach
      5. Podniesienie wartości czujników o 2.
      6. Oczekiwany rezultat - alarm odwołany (w całości instalacji)
   2. Scenariusz 2:
      1. Skrypt uruchamia instalacje z zadaną ilością czujników.
      2. Dla wszystkich ustawiania jest wartość alarmowa 5.
      3. Skrypt wyłącza centrale startową i uruchamia zapasową.
      4. Za pomocą skryptu na kolejnych czujnikach przekraczany jest próg alarmowy o 1.
      5. Oczekiwany rezultat - alarm na wszystkich czujnikach
   3. Scenariusz 3:
      1. Skrypt uruchamia instalacje z zadaną ilością czujników.
      2. Dla wszystkich ustawiania jest wartość alarmowa 5.
      3. W module przeglądarkowym (oraz w skrypcie bash) wpisujemy niepoprawne dane (zamiast liczby inne znaki itp.)
      4. Oczekiwane działanie - aplikacja odporna na niepoprawne danych
3. **Podział prac w zespole.**
   1. Jakub Dudziak
      1. Dokumentacja wstępna.
      2. Implementacja czujników.
      3. Testowanie funkcjonalne.
      4. Testowanie poszczególnych modułów.
   2. Paweł Joński
      1. Implementacja centrali.
      2. Centrala zapasowa.
      3. Testowanie jednostkowe centrali.
   3. Michał Sacharczuk
      1. Dokumentacja wstępna.
      2. Implementacja czujników.
      3. Skrypt powłoki.
      4. Testowanie jednostkowe czujników.
   4. Michał Tomaszewski
      1. Dokumentacja końcowa.

### Moduł do Wireshark.

* + 1. Przeglądarka.
    2. Implementacja centrali.

1. **Harmonogram prac (minimum 2 punkty kontrolne dla odbioru częściowych funkcji/modułów projektu).**
   1. 8.05
      1. Parsowanie komunikatów przez wireshark.
      2. Połączenie czujników z centralą (zestawiona komunikacja, obsługa wielu czujników na raz, logika alarmu)
   2. 22.05
      1. Moduł przeglądarkowy.
      2. Skrypt powłoki
2. **Adres projektu na serwerze kontroli wersji.**
   1. https://github.com/Sacharc/TIN