

Wybrane aspekty implementacji aplikacji do sterowania zasilaczami 200A z interfejsami Pico

wersja 1.0

1. Struktura programu i przepływ danych

Przyjęto koncepcję, że aplikacja będzie pracować w dwóch trybach: lokalnym i zdalnym, w celu skrócenia czasu potrzebnego na testowanie aplikacji oraz pielęgnację kodu. Takie założenie powoduje, że niektóre moduły oprogramowania są nieaktywne w zależności od trybu pracy. Diagramy 1 i 2 pokazują to w uproszczony sposób.

Drugim założeniem, które wpływa na strukturę programu, jest możliwość wyboru sterowania lokalnego albo zdalnego. W zależności od wyboru miejsca sterowania, zmienia się przepływ danych pomiędzy modułami.

Data flow diagram in case of local control

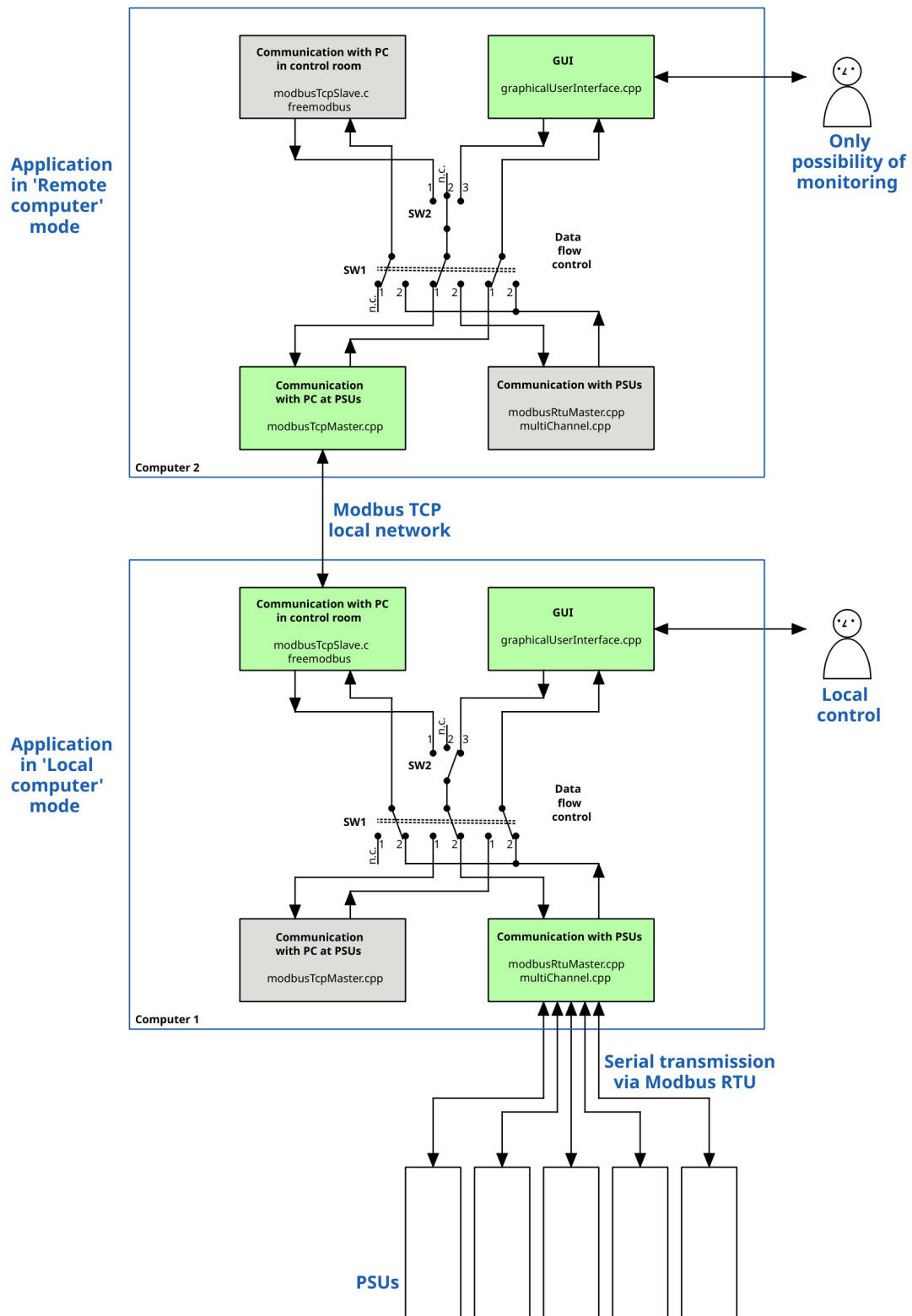


Diagram 1.

Data flow diagram in case of remote control

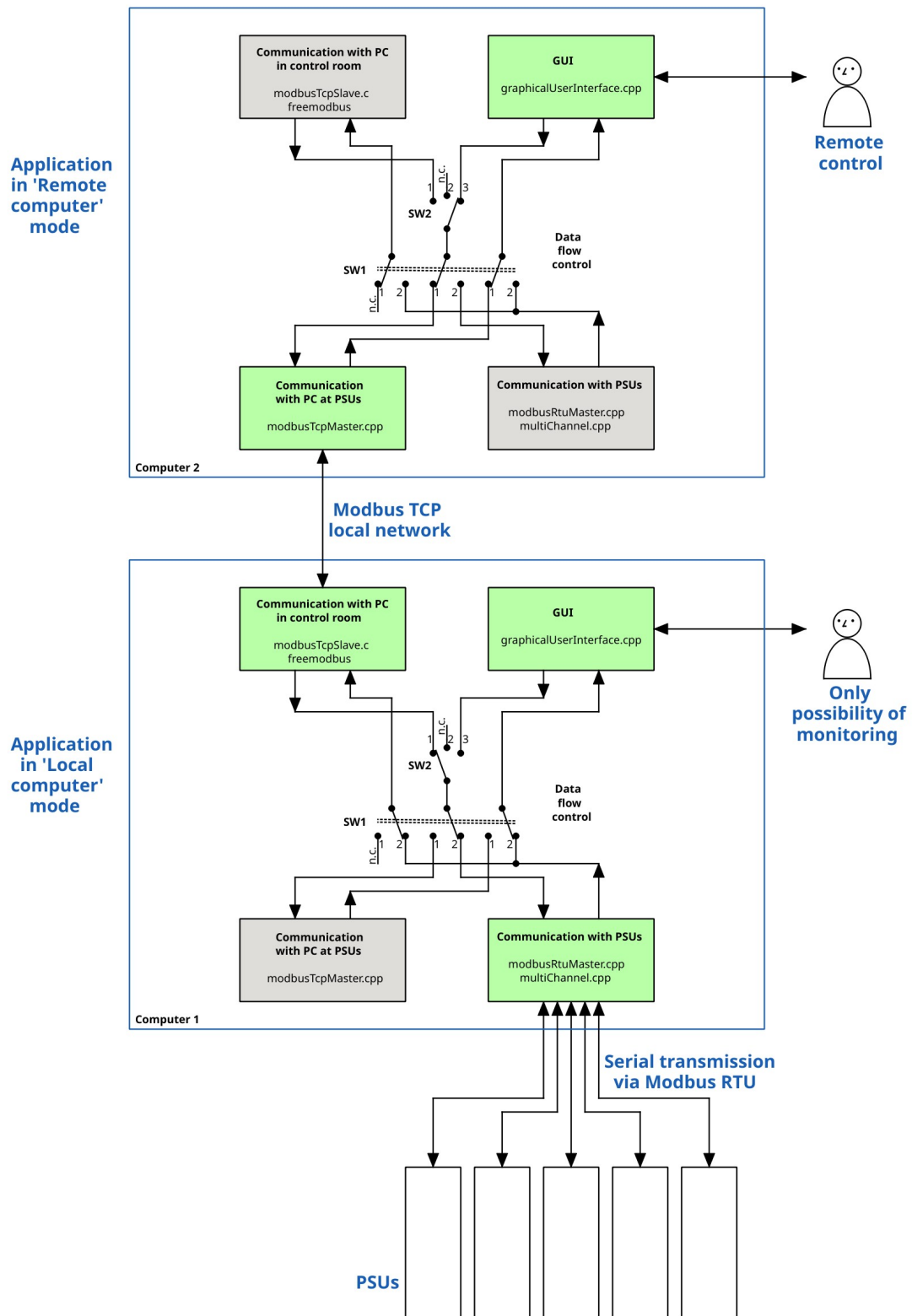


Diagram 2.

2. Wątki

Wątki pokazuje uproszczony diagram 3. W aplikacji mogą działać 3 wątki:

- wątek główny FLTK, który obsługuje GUI i uruchamia wątek peryferyjny;
- wątek peryferyjny wykonuje następujące funkcje:
 - odczytuje plik konfiguracyjny;
 - w trybie lokalnym uruchamia trzeci wątek: wątek serwera (slave'a) Modbusa TCP;
 - w pętli cyklicznie obsługuje warstwę niższą; w zależności od trybu pracy, albo prowadzi komunikację z zasilaczami przez Modbusa RTU w trybie lokalnym, albo prowadzi komunikację z serwerem (slave'm) Modbusa TCP w trybie zdalnym;
- wątek serwera (slave'a) Modbusa TCP, który jest aktywny tylko w trybie lokalnym.

Threads running in the application and data sharing

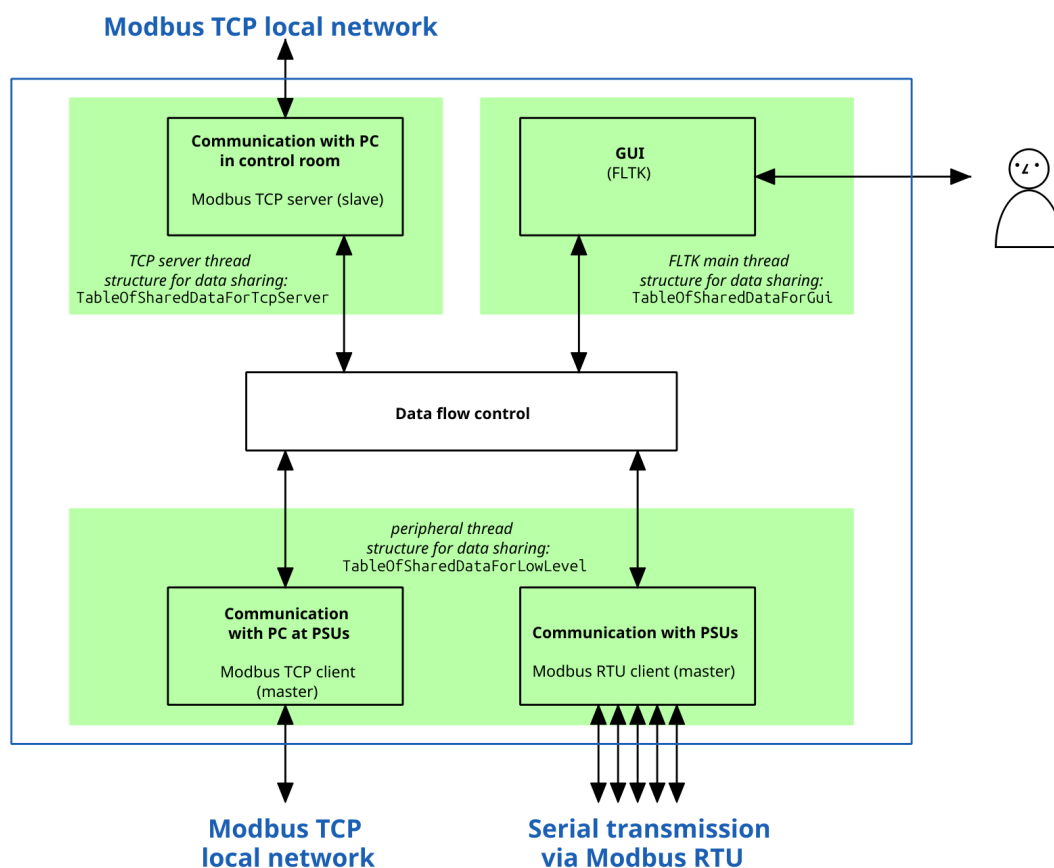


Diagram 3.

Do wymiany danych pomiędzy wątkami służy klasa `DataSharingInterface`, która zawiera komplet zmiennych współdzielonych, odnoszących się do pojedynczego zasilacza.