# ModBus RTU

<https://pl.wikipedia.org/wiki/Modbus>

Wcześniejsze zajęcia dotyczyły komunikacji ASCII, był to UART bardzo podobny do RS232, który na Lab02 zrealizowaliśmy też na magistrali RS485.

Szczegóły zostały przedstawione na wykładzie, teraz interesuje nas użycie UART w MCU do wymiany danych po *magistrali* RS485, i to od razu RTU, a nie ASCII.

Aby nie było zbyt łatwo, zajmujemy się jednym z protokołów, którego używają zarówno automatycy jak i embedowcy. Jedni i drudzy go nienawidzą. Automatycy za to, że przypomina programowanie MCU, a embedowcy za to, że jest toporny jak większość mechanizmów wymiany danych używanych w PLC.

Protokół jest prosty i dobrze udokumentowany, ale jako że leży na granicy dwóch zupełnie różnych światów, to nie doczekał się dobrych standardowych implementacji w C, C++, C#, Python. Jest mnóstwo przykładów, są dziesiątki bibliotek, ale te dobre są złożone (np. szablony C++ i zaawansowane OOP), a te proste mają wbudowane podstawowe błędy. Część nawet prostych narzędzi jest płatna.

Dobrym treningiem będzie użycie <https://github.com/debevv/nanoMODBUS> w projekcie na Pico, którego zadaniem będzie cykliczne przełączanie przekaźnika (razem jest ich 8):

<https://www.waveshare.com/wiki/Protocol_Manual_of_Modbus_RTU_Relay>

Moduł RTU jest skonfigurowany domyślnie: 9600,8N1,id=1.

1. Przeanalizuj dokumentację modułu przekaźnikowego RTU. Wyznacz ramkę przełączającą przekaźnik przypisany do twojego stanowiska: 0..7 i użyj mobilnego RTU, aby sprawdzić poprawność działania kodu.
2. Dołącz do swojego projektu bibliotekę nanomodbus (github) i zademonstruj jej działanie, przełączając przekaźniki 0..7 co 1 s
3. Wzorcowy RTU w labie odpowiada na zapytanie o zawartość rejestru 0x1234 zwracając 16-bitową liczbę będącą czasem w sekundach do końca zajęć. Wyświetl tę liczbę w konsoli.
4. Zaimplementuj modbus slave na swoim Pico, wystawiając do odczytu Holding Register o numerze będącym 4 ostatnimi cyframi swojego albumu, a w tym rejestrze zapisz ocenę jaką chcesz uzyskać z kursu w postaci float32. Wskazówka: rejestry modbus mają 16 bitów, jak to rozwiązać?
5. Przygotuj program który sprawdza, jaki baudrate oraz jaki slave\_id ma niewiadomy moduł dostarczony przez prowadzącego zajęcia.
6. \*Wyświetl bieżącą moc, prąd i napięcie z miernika energii PZEM-004T.