



UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA U
NOVOM SADU



Milan Kostić

Stefan Karać

Simulacija elektroenergetskih sistema sa kritičnom misijom

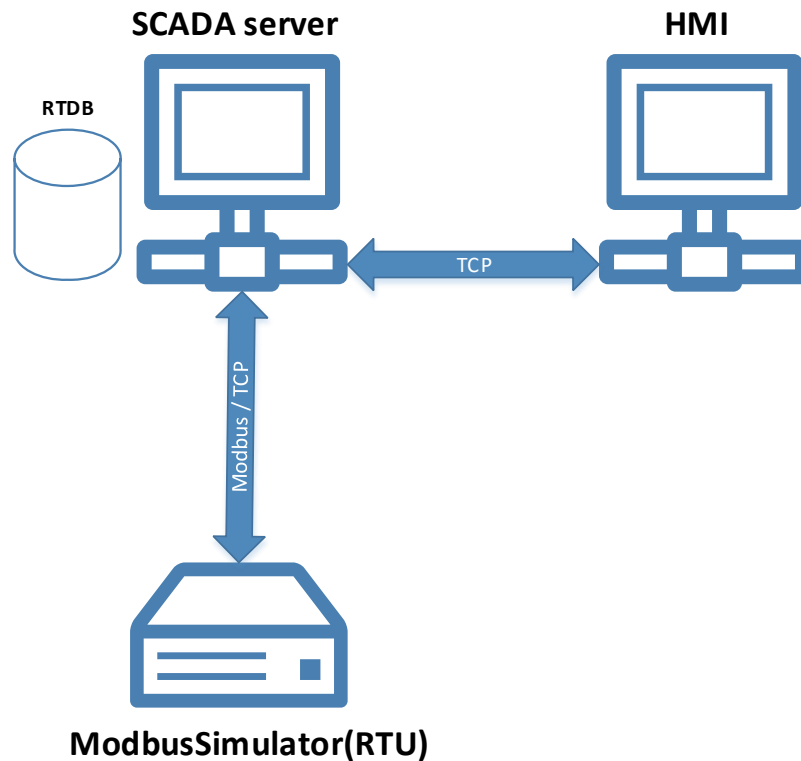
Predmetni projekat

Profesor dr. Vladimir Pavlica

Novi Sad, 2016.

1. Arhitektura sistema

Zadatak opisanog softverskog rješenja je simulacija SCADA sistema koji upravlja mješalicom betona. Arhitektura sistema u globalu (Slika 1.) sastoji se od tri modula, SCADA server, *Remote* uređaj (*Modbus Simulator*) i HMI modul (*Human Machine Interface*). Modbus Simulator sadrži adrese digitalnih i analognih ulaza i izlaza koji opisuju mješalicu. Pomenuti SCADA sistem prikuplja podatke sa Modbus Simulatora na svake dvije sekunde, vrši obradu i čuva ih u radnoj memoriji. HMI je modul namjenjen korisniku, ima ulogu da prikaže trenutno stanje sistema kao i da šalje komande SCADA serveru na zahtjev korisnika.

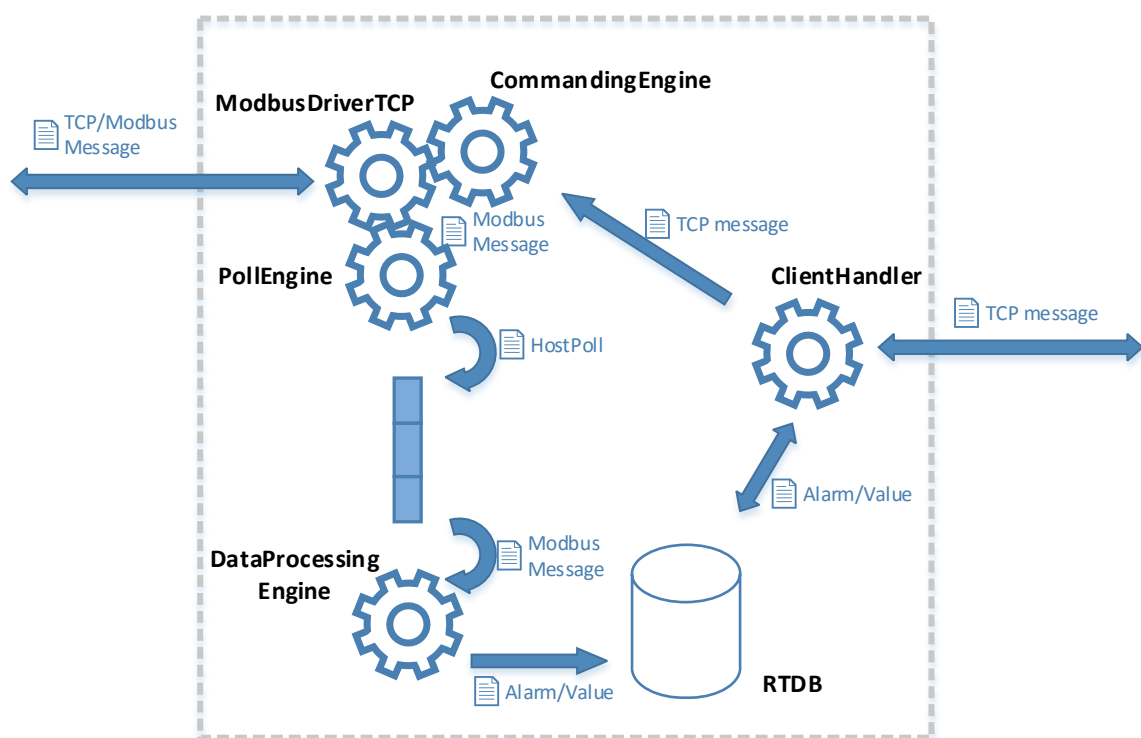


Slika 1. Arhitektura simuliranog SCADA sistema

U ovom rješenju ulogu Remote uređaja ima aplikacija Modbus Simulator koja komunicira sa SCADA serverom putem Modbus/TCP protokola preko porta 502. HMI predstavlja konzolnu aplikaciju koja komunicira sa SCADA serverom putem TCP protokola. Korisnik ove aplikacije ima mogućnost pregleda trenutnih vrijednosti posmatranog sistema kao i zadavanje komande za pražnjenje mješalice. Takođe ovaj modul ima dužnost da korisnika obavijesti ukoliko se desi alarm koji se generiše u okviru SCADA servera. Alarm se generiše kada

je detektovano pražnjenje mješalice a dozvola pražnjenja nije izdata. SCADA server je od najvećeg značaja za opisano rješenje i o njemu će biti detaljno rečeno u nastavku dokumenta.

Arhitektura SCADA servera (Slika 2.) sastavljena je od šest jedinstvenih komponenti. Logički, SCADA server se može podijeliti na dvije cjeline. Prva cjelina je okrenuta klijentskoj strani i zadužena je za generisanje komandi i pristup *real-time* bazi, kao i slanje poruke o generisanju alarma od strane SCADA servera prema klijentu. Ulaz i izlaz ove cjeline su TCP poruke. Druga cjelina orijentisana je prema Remote uređaju, tj. Modbus Simulatoru i zadužena je da šalje i prima Modbus poruke putem TCP protokola, vrši obradu podataka i smiješta ih u bazu.



Slika 2. Arhitektura SCADA servera

ModbusDriverTCP je objekat namjenjen za slanje Modbus poruka kao i za primanje istih. Pošto ovaj modul vrši komunikaciju sa Modbus Simulatorom putem TCP/Modbus protokola, nakon što pošalje poruku, ModbusDriverTCP očekuje poruku koju će dobiti kao odgovor od simulatora. Njegova uloga je i da od dobijenog niza bajtova generiše odgovarajuću Modbus poruku.

PollEngine je objekat čija je uloga da na svake dvije sekunde vrši *polling* a zatim uređeni par poslate poruke i odgovora smiješta u uređenu strukturu u vidu HostPoll objekata.

DataProcessingEngine je objekat koji vrši *crunching*. Njegova uloga je da iz strukture u koju PollEngine upisuje poruke, čita jednu po jednu i vrši njenu obradu koja se završava upisom u bazu podataka.

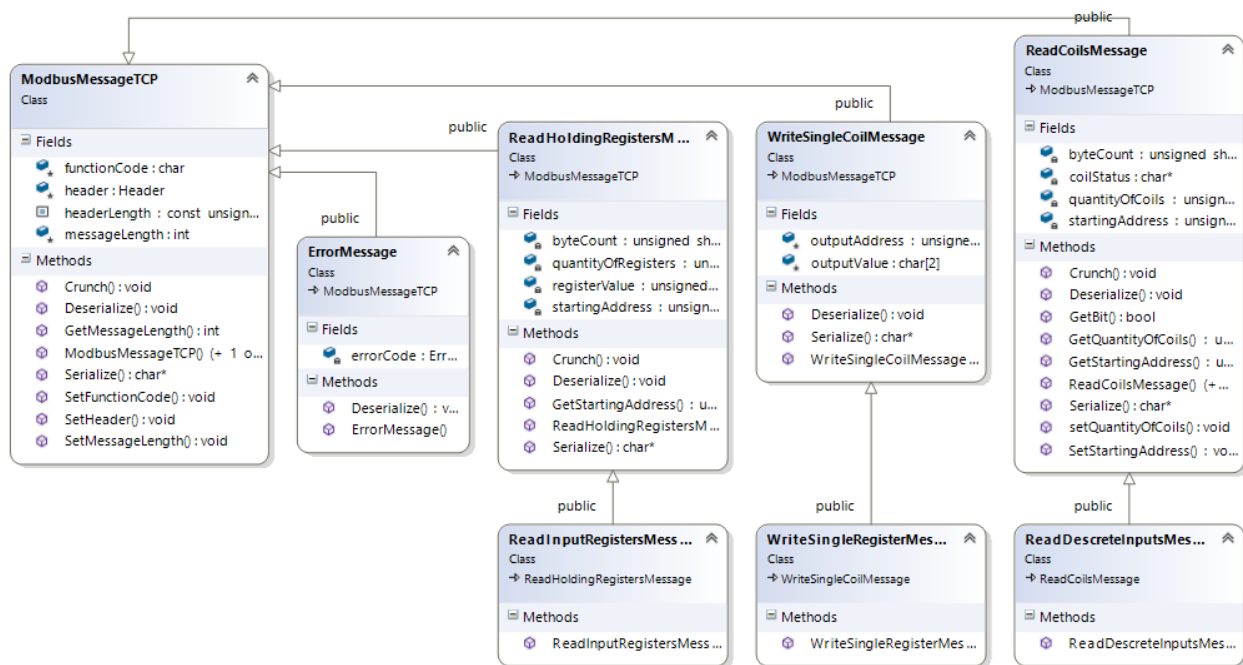
RTDB je objekat koji simulira *real-time* bazu podataka. Ovaj objekat nema nikakvu funkcionalnost osim posjedovanja trenutnih vrijednosti mjerenih veličina i alarma.

ClientHandler je objekat namjenjen komunikaciji sa klijentom. ClientHandler otvara novi *socket* za svakog novog klijenta koji pošalje zahtjev, a zatim prima i procesira TCP poruke. ClientHandler se obraća RTDB objektu u slučaju zahtjeva za prikaz trenutnih vrijednosti iz baze ili CommandingEngine objektu u slučaju korisničkog zahtjeva za komandovanje. Paralelno sa ovim, ClientHandler provjerava da li se desio alarm i šalje poruku korisniku u tom slučaju.

CommandingEngine je objekat čija je uloga da od dobijenog niza bajta kojeg dobije od ClientHandler-a generiše odgovarajuću Modbus poruku i prosledi je ModbusDriverTCP-u na slanje. Pored toga, CommandingEngine na svaku sekundu vrši slanje poruka za postavljanje novih vrijednosti određenih analognih izlaza (količina vode, pijeska i šljunka).

2. Dizajn sistema

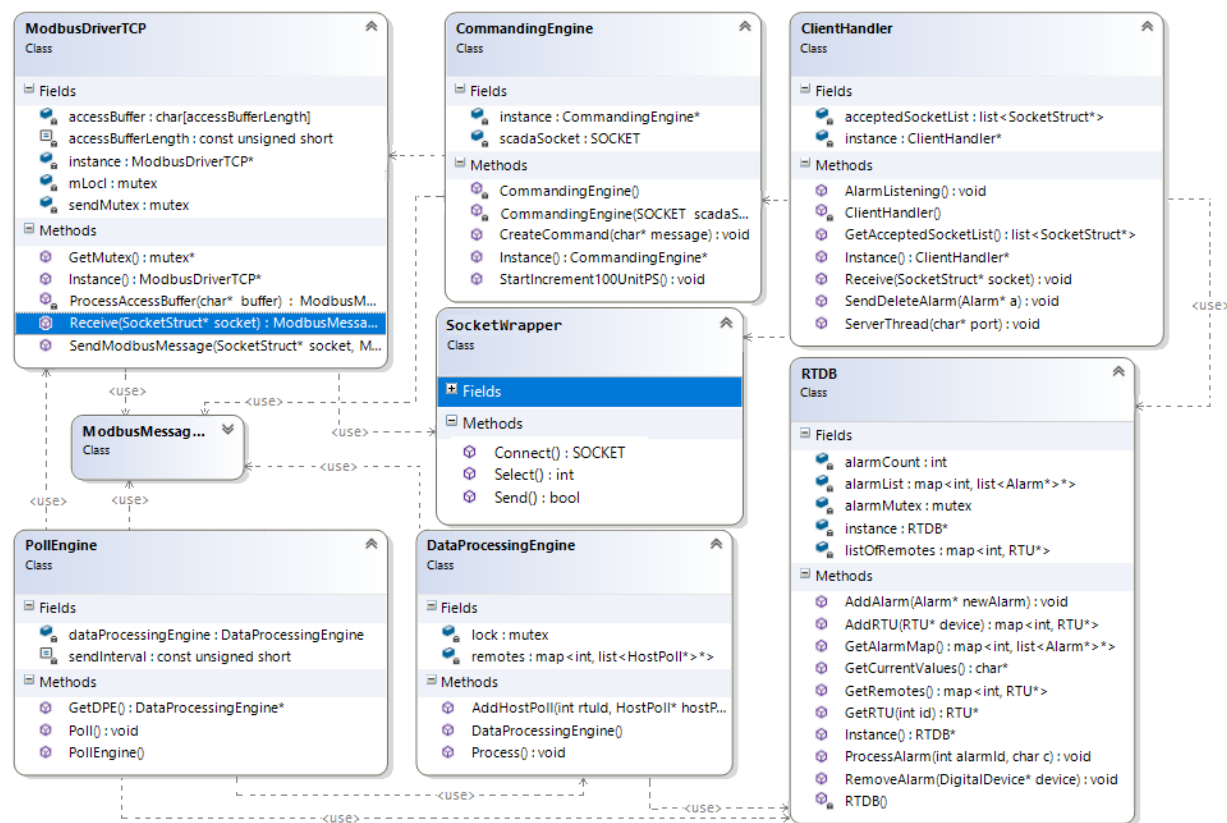
U ovom poglavlju biće opisan dizajn SCADA servera kroz klasne i sekvencijalne dijagrame. Klasni dijagram (Slika 3.) pokazuje međusobnu zavisnost klasa koje opisuju Modbus poruke.



Slika 3. Klasni dijagram Modbus poruka

Pošto svaka Modbus poruka ima raziličitu strukturu, svaku poruku opisuje posebna klasa s tim da su zajedničke osobine i metode smještene u klasu koju nasleđuju, ModbusMessageTCP. Poruke koje su se koristile u ovom zadatku opisane su klasama tako da svaka ima svoju metodu za odradu, deserijalizaciju, serijalizaciju i slično. Ovim postupkom izbjegnuto je gomilanje programskog koda u metodama ModbusDriverTCP-a i DataProcessingEngine-a, pa se nad svakom porukom poziva *Crunch()*, *Serialize()*, *Deserialize()*...

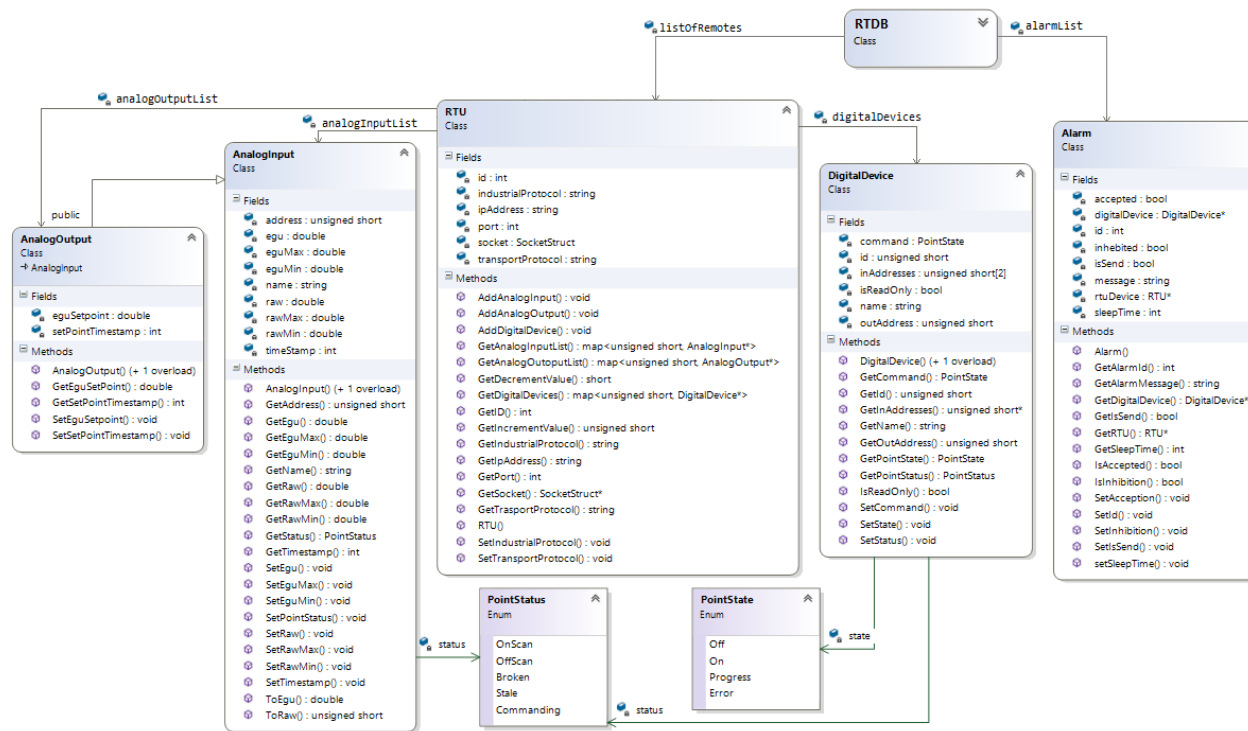
Klasni dijagram (Slika 4.) sadrži klase funkcionalnih modula SCADA servera. Svaka klasa sa klasnog dijagrama odgovara jednoj instanci u sistemu, pa su klase realizovane kao *Singleton* objekti. Objekti nemaju međusobnu zavisnost ali koriste jedan drugog po scenariju navedenom u prethodnom poglavlju.



Slika 4. Klasni dijagram funkcionalnih modula SCADA servera

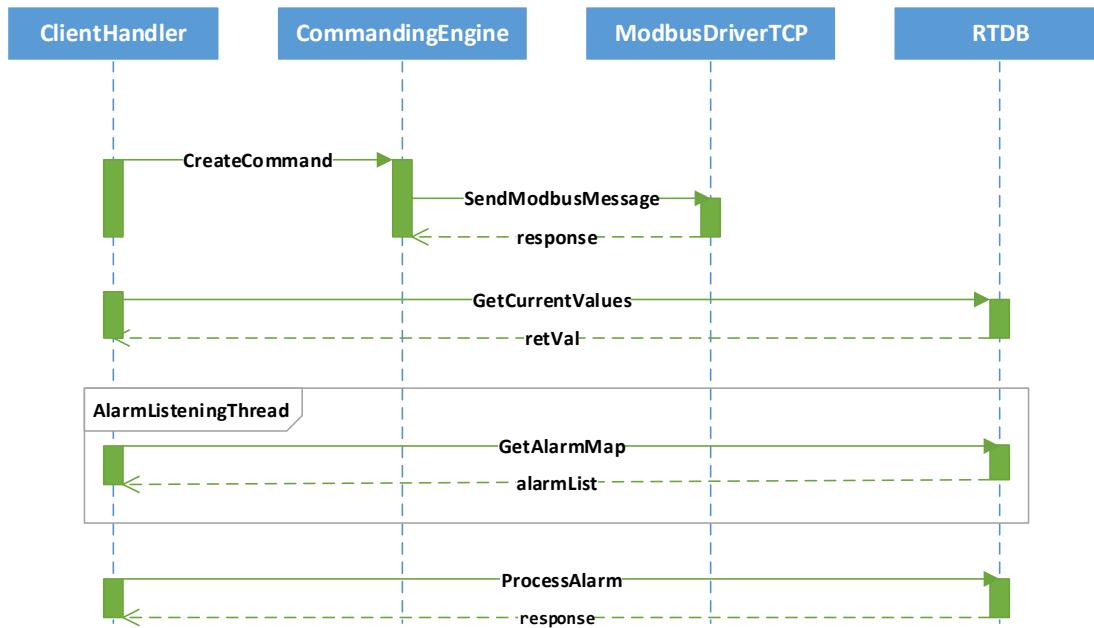
Klasni dijagram (Slika 5.) opisuje model podataka koji se nalazi u *real-time* bazi. Objekat klase RTDB sadrži kolekciju RTU uređaja i kolekciju alarma za svaki RTU uređaj. U ovom zadatku uvijek će biti jedan uređaj i jedan alarm jer je tako navedeno u postavci. Uz model SCADA sistema, uvedene su dvije strukture PointState i PointStatus. PointState predstavlja stanje digitalne veličine a PointStatus predstavlja status posmatrane veličine (da li se posmatra dati ulaz/izlaz). RTU uređaj sadrži kolekciju objekata AnalogInput, AnalogOutput i DigitalDevice koji predstavljaju vrijednosti analognih ulaza, izlaza i digitalnih uređaja. DigitalDevice je klasa koja

opisuje digitalni uređaj sa jednom izlaznom adresom u dvije ulazne adrese. Dvije ulazne adrese koriste se zbog toga što stanje uređaja ima četiri vrijednosti (PointState). Alarm je objekat koji predstavlja model alarma u realnom sistemu. Sadrži referencu na RTU uređaj kao i na digitalni uređaj na kojem se desio događaj.



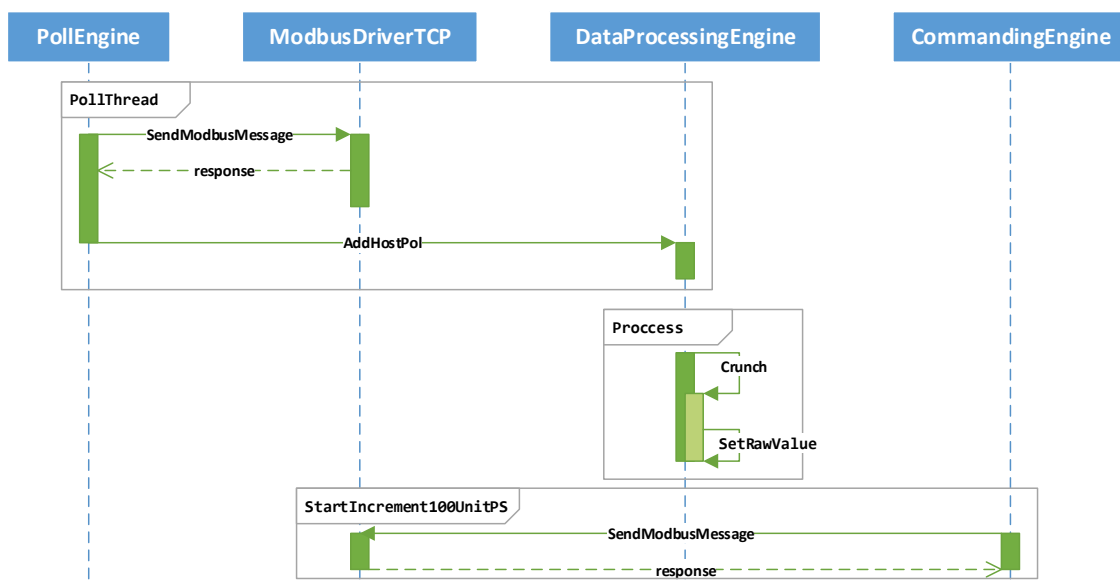
Slika 5. Model podataka sistema

Sekvencijalni dijagram (Slika 6.) prikazuje tok saradnje klijentske aplikacije sa SCADA serverom. Na njemu su prikazani sekvencijalni pozivi metoda modula za kreiranje komandi i pristup bazi podataka.



Slika 6. Tok saradnje klijentske aplikacije sa SCADA serverom

Sekvencijalni dijagram (Slika 7.) prikazuje tok saradnje SCADA servera sa Modbus Simulatorom. Sadrži sekvencu poziva metoda koje primaju poruke sa simulatora, obrađuju i smiještaju u bazu podataka, kao i sekvencu metoda koje vrši povremeno slanje poruka na simulator.



Slika 7. Tok saradnje SCADA servera sa simulatorom