



UNIVERZITET U NOVOM SADU  
FAKULTET TEHNIČKIH NAUKA U  
NOVOM SADU



Fakultet Tehničkih nauka

# Simulacija elektroenergetskih sistema sa kritičnom misijom

Regulacija temperature - Dokumentacija

Mentori:

Vladimir Nikolić  
Vladimir Pavlica

Autori:

Mihailo Vasiljevic  
Davor Nađ

Novi Sad, 2017

# 1.Sistem

U ovom odeljku sadrži kratak opis projektnog zadatka, kao i dizajn sistema.

## 1.1 Projektni zadatak

Potrebno je projektovati i implementirati SCADA sistem koji prikuplja podatke od Modbus TCP simulatora na svakih 2s. Nakon prikupljanja podataka treba da se podaci obrade i po potrebi da se korisniku saopšte alarmna stanja. Sistem treba da omogući i izdavanje komandi.

Opis kontrolisanog procesa:

Nadzire se i upravlja se temperaturom u prostoriji.

Monitorisane veličine:

1. Spoljašnja temperature (analog input)
2. Unutrašnja temperature (analog input)
3. Stanje grejača (digital input, on-off-error)
4. Zadata temperature (analog input)

Kontrolisane veličine:

Closed loop:

Shodno vrednostima monitorisanih veličina treba da se za postizanje zadate unutrašnje temperature (set-point) koja je veća od spoljašnje, uključi grejač I drži uključenim dok trenutna unutrašnja temperatura ne dostigne ciljanu vrednost.

1. Stanje grejača (digital output, on-off)

Alarmna stanja:

Sistem je u alarmnom stanju ukoliko istekne 15 sekundi od izdavanja komande za promenu stanja grejača, a stanje grejača se ne promeni ili stanje grejača postane "error".

Alarm mora biti vidljiv na HMI-u i posebno istaknut. Po potvrdi alarma od strane korisnika alarm se može ukloniti tek nakon prestanka razloga koji je doveo do aktivacije alarma.

HMI:

Korisnička sprega SCADA sistema je klijentska aplikacija za SCADA server, a treba da omogući korisniku da bude u mogućnosti da komanduje kontrolisanim uređajima, kao i da ima uvid u trenutno stanje sistema. Klijentska aplikacija komunicira sa SCADA serverom putem TCP/IP protokola na unapred definisanom portu.

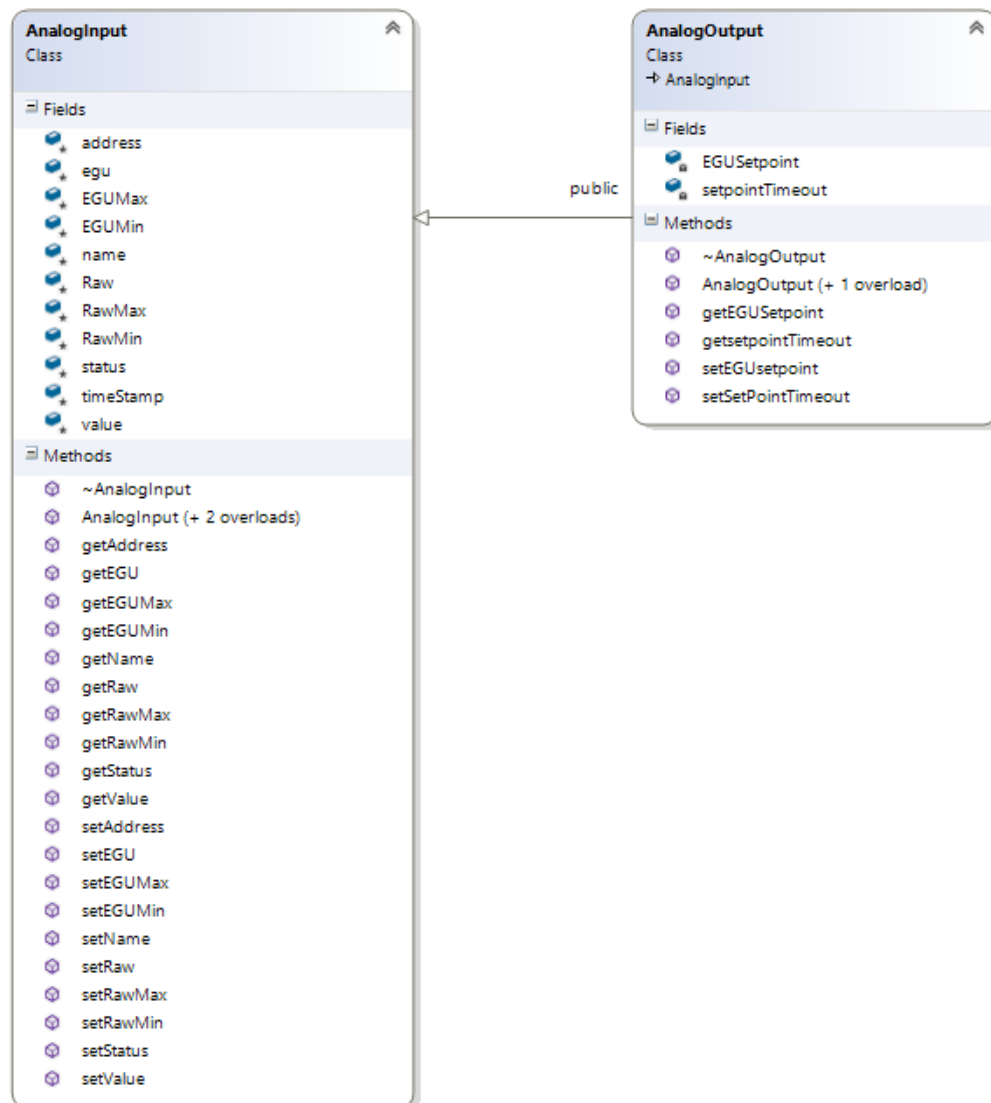
Napomena:

Konfiguracija upravljačkog bloka treba da sadrži podatke za sva polja iz SCADA modela.

SCADA server se konfiguriše iz datoteke.

## 1.2 Dijagram klasa

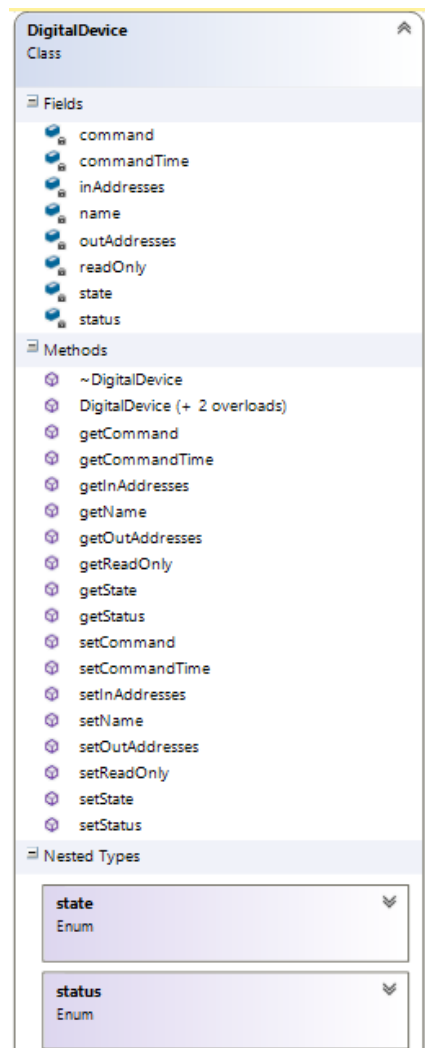
Dijagram klasa je vrsta strukturnog dijagrama u softverskom inženjeringu, koji opisuje strukturu sistema predstavljajući klase unutar sistema, kao i njihove atribute i odnose.



Slika 1.1 Analog input i analog output klase

Model podataka ovog SCADA sistema čine klase *Analog input*, *Analog output*, *Digital Device*, *Remote telemetry unit*, *alarm* i *engineering unit*.

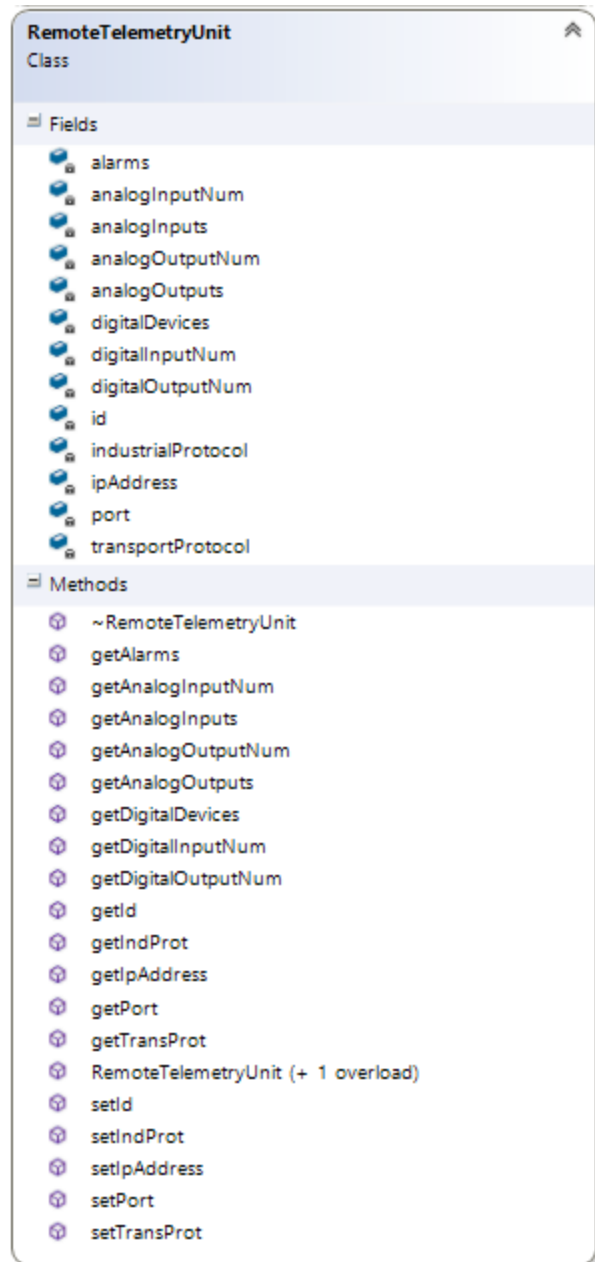
Na slici 1.1 su prikazane klase *Analog input* i *analog output*. Klasa *analog input* sadrži polja *address* (adresa analognog ulaza), *egu* (inženjerska jedinica), *EGUMin* (minimalna vrednost inž. jedinice), *EGUMax* (maksimalna vrednost inž. jedinice), *name* (ime analognog ulaza), *Raw* (vrednost dobijena od simulatora), *RawMin*, *RawMax*, *status*, *timeStamp* i *value* (konvertovana raw vrednost u inž. jedinice), kao i getere i setere za ova polja. Klasa *analog output* nasledjuje klasu *analog input* i poseduje još polja *EGUSetpoint* i *setpointTimeout*, kao i getere i setere za ova polja.



Slika 1.2 DigitalDevice klasa

Na slici 1.2 je prikazana *DigitalDevice* klasa. Ova klasa sadrži polja *name* (ime digitalnog uređaja), *readOnly*, *inAddresses* (ulazne adrese), *outAddresses* (izlazne adrese), *state* (stanje koje je enumeracija i može biti *ON*, *OFF*, *TRANSIENT* ili *ERROR*), *status* (status koji je enumeracija i može biti *INPROGRESS* ili *FINISHED*), *command* (zadata komanda) i *commandTime* (vreme kada je komanda zadata).

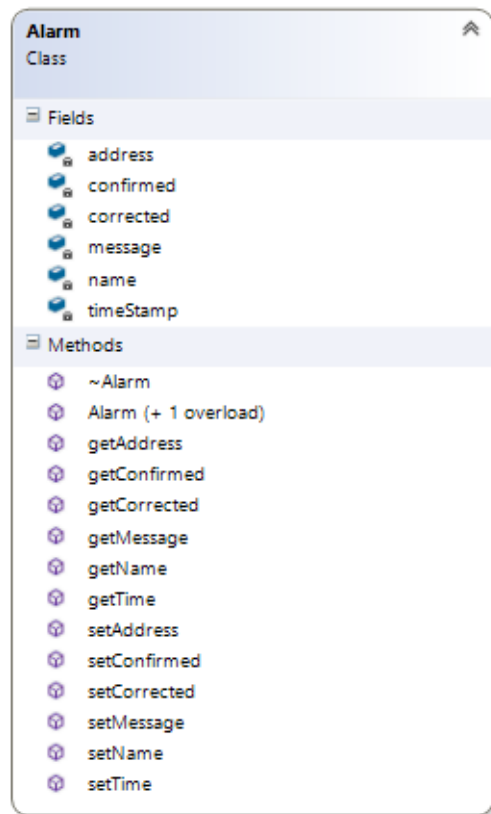
Na slici 1.3 je prikazana *RemoteTelemetryUnit* klasa. Ova klasa sadrži polja *id* (id digitalnog uređaja), *industrialProtocol*, *transportProtocol* (protokol za transport), *ipAddress*, *port*, *analogInputs* (analogni ulazi nakačeni na rtu), *analogInputNum* (broj analognih ulaza), *analogOutputs* (analogni izlazi nakačeni na rtu), *analogOutputsNum* (broj analognih izlaza), *digitalDevices* (digitalni uređaji nakačeni na rtu), *digitalInputNum* (broj digitalnih ulaza), *digitalOutputNum* (broj digitalnih izlaza), *alarms* (alarmi nakačeni na rtu).



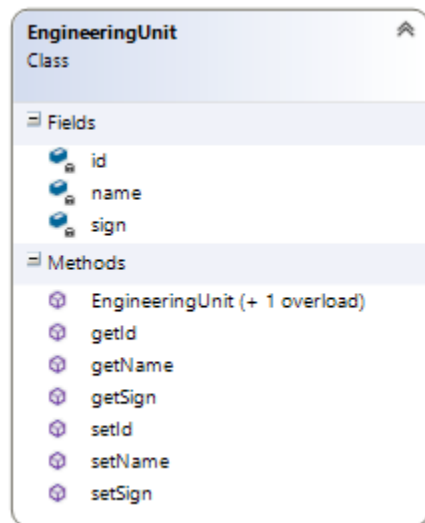
Slika 1.3 RemoteTelemetryUnit klasa

Na slici 1.4 je prikazana klasa *Alarm*. Ova klasa sadrži polja *name* (ime alarma), *address* (adresa alarma), *message* (poruka alarma), *corrected* (podatak da li je alarm ispravljen), *confirmed* (podatak da li je alarm potvrđen), *timeStamp* (vreme kada se generisao alarm).

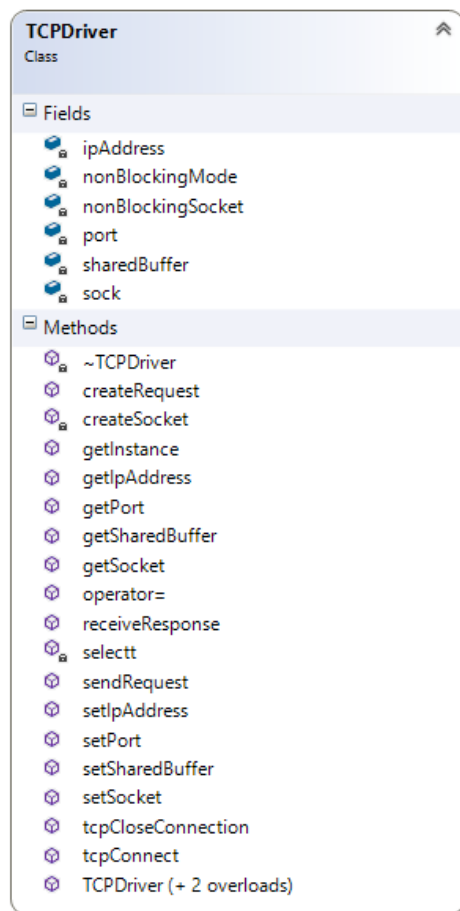
Na slici 1.5 je prikazana klasa *EngineeringUnit*. Ova klasa sadrži polja *id*, *name* (ime egu-a) i *sign* (oznaka za egu), kao i getere i setere za ova polja.



Slika 1.4 Klasa alarm

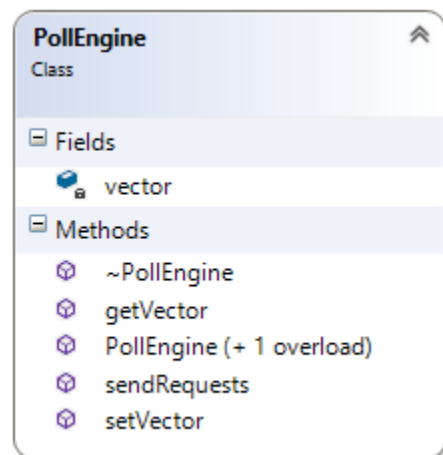


Slika 1.5 Engineering unit klasa



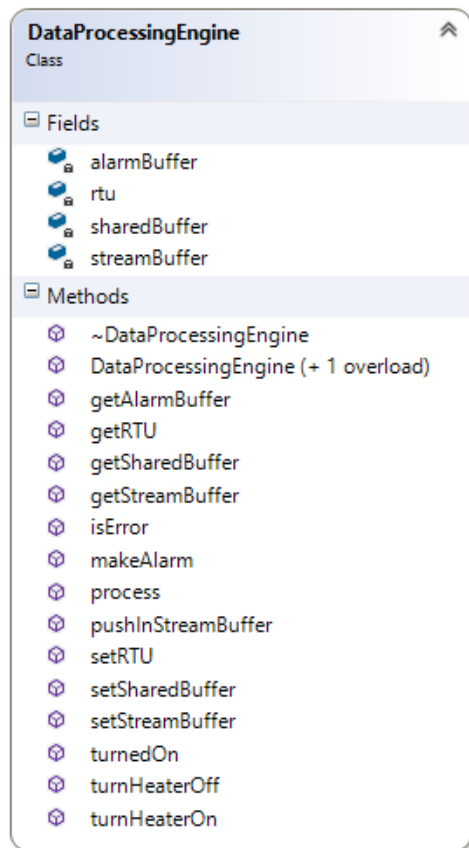
Slika 1.6 TCPDriver klasa

Na slici 1.6 je prikazana TCPDriver klasa. Ova klasa implementira Modbus TCP protokol i nudi API za slanje komande ka TCP master uređaju i poll podataka. Ova klasa sadrži metode za kreiranje zahteva, kreiranje soketa, pravljenje tcp konekcije, zatvaranje tcp konekcije, slanje zahteva i primanje odgovora, kao i getere i setere za njena polja.



Slika 1.7 PollEngine klasa

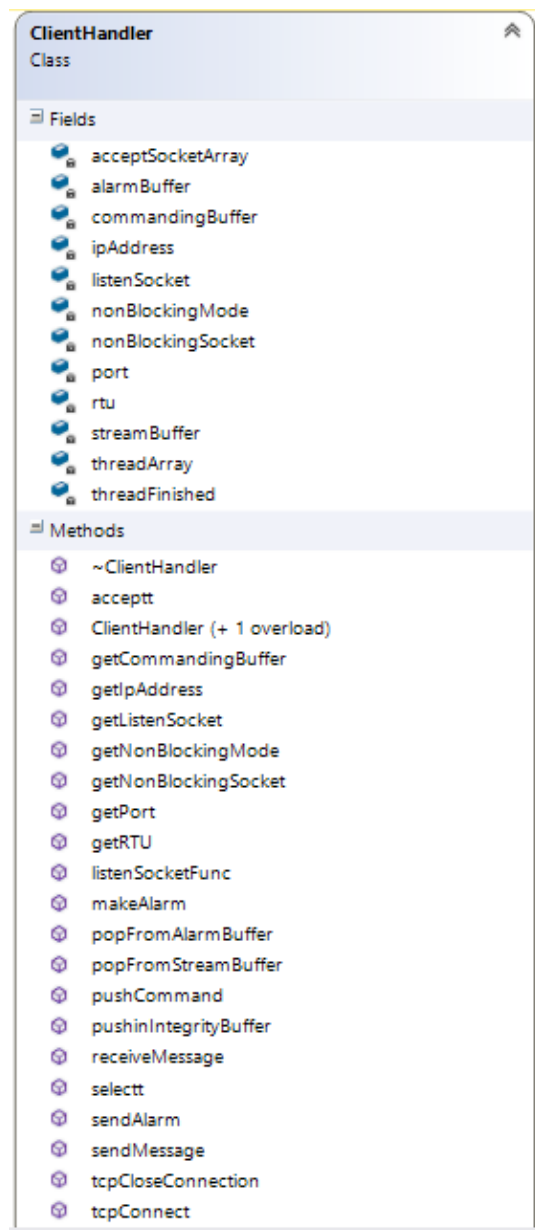
Na slici 1.7 je prikazana klasa *PollEngine*. Ova klasa vrši periodično poll svih podataka (na svakih 2s) i prihvaćene podatke smešta u interne bafere. Nakon toga *data processing engine* obrađuje podatke.



Slika 1.8 Data processing engine klasa

Data processing engine sadrži polja *rtu*, *alarmBuffer* (buffer u koji se smeštaju podaci vezani za alarm kada se generiše alarm), *sharedBuffer* (buffer u koji se smeštaju podaci vezani za analogne ulaze, analogne izlaze, digitalne izlaze i digitalne uređaje), *streamBuffer*. *Shared buffer* se koristi za komunikaciju između poll engine-a i data processing engine-a. U ovaj buffer se smeštaju podaci dobijeni od Modbus simulatora i kasnije se ti podaci skidaju sa ovog buffera u data processing engine-u i obrađuju. Stream buffer sadrži podatke vezane za analogne ulaze, analogne izlaze, digitalne uređaje i on se šalje klijentu koji obrađuje ove podatke i prikazuje na ekranu. Data processing klasa sadrži metode za procesiranje podataka, proveravanje alarmnih stanja, pravljenje alarma, uključivanje ili isključivanje grejača itd. Ova klasa je prikazana na slici 1.8.





Slika 1.9 ClientHandler klasa

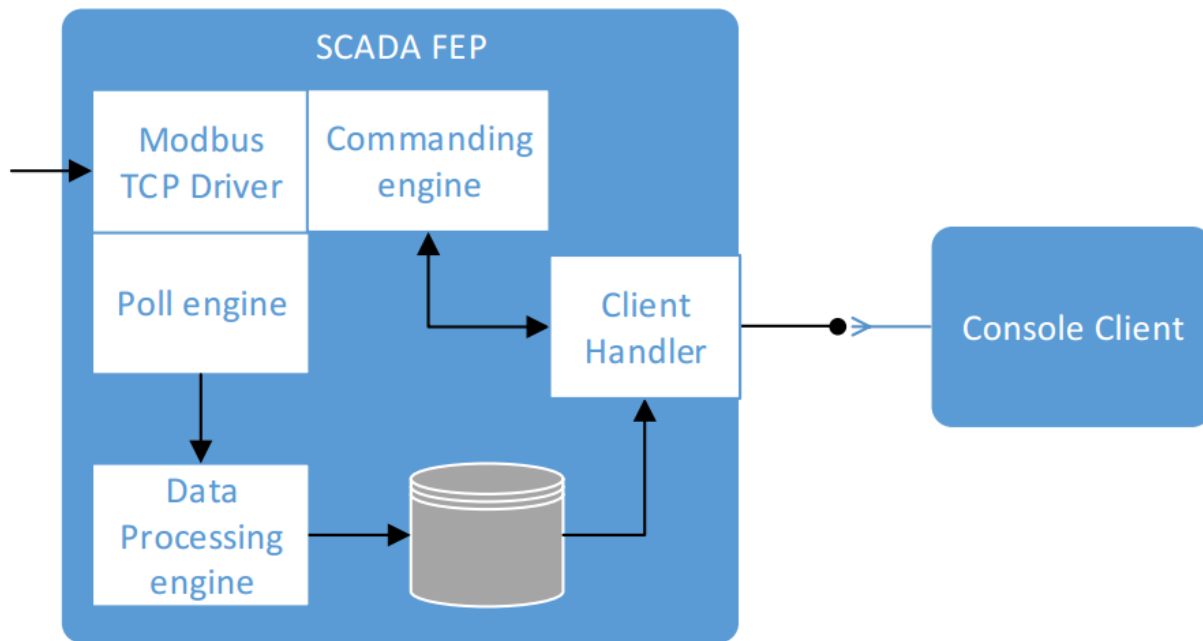
*ClientHandler* klasa prihvata zahteve od eksternog sistema, tj. klijenta, obrađuje ih i vraća odgovor. Takođe je podržan rad sa više klijenata. Ova klasa sadrži metode za uspostavljanje konekcije sa klijentima, zatvaranje konekcije, slanje poruka, ili alarma klijentima, prihvatanje poruke od klijenta (koje se kasnije obrađuju i po potrebi šalju odgovori klijentu), itd. Na slici 1.9 je prikazana klasa *ClientHandler*.

*ConsoleClient* predstavlja poseban projekat i klasu koja omogućuje tekstualni ispis monitorisanih veličina (analognih ulaza(unutrašnja, spoljašnja i zadata temperatura), i digitalnih uređaja (stanje grejača)), kao i alarma na konzolu. U svakom trenutku na konzoli su prikazana trenutna stanja temperatura (analognih ulaza), stanje grejača (digitalni uređaj) i alarmi.

### 1.3 Arhitektura sistema

Arhitektura SCADA sistema je prikazana na slici 1.10. Ovaj sistem sadrži sledeće komponente:

- Modbus TCP Driver
- Poll engine
- Data Processing engine
- Commanding engine
- Client Handler
- Model
- Console Client



Slika 1.10 SCADA sistem

**Modbus TCP Driver** implementira Modbus TCP protokol i nudi API za slanje komande ka TCP master uređaju i poll podataka.

**Poll engine** periodično vrši poll svih podataka, i prihvaćene podatke smešta u interne bafere. Takođe javlja processing engine komponenti da su podaci spremni za obradu.

**Data processing engine** vrši sekundarnu obradu primljenih podataka koje čita iz privremenih prihvatnih bafera nakon završenog poll ciklusa. Nakon sekvencijalne obrade, vrednost se upisuje u internoj bazi podataka.

**Commanding engine** prima komande od Client Handler komponente i šalje je na izvršavanje.

**Client Handler** komponenta prihvata zahteve od eksternog sistema, u našem slučaju klijenta, obrađuje ih i vraća odgovor. Takođe je potrebno podržati rad sa više klijenata.

**Console Client** aplikacija omogućava tekstualni ispis potrebnih elemenata.