

## Õppeaine 6: SIDELIIDESED JA -PROTOKOLLID [M006ET]

### Peatükk 6. MODBUS-protokoll

#### 6.1 MODBUS protokoll ülevaade

Protokoll MODBUS on enamasti keerdpaarkaablil töötav protokoll, mille puhul eristatakse kahte peamist edastusmoodust ehk protokoll alamliiki: ASCII või RTU (*remote terminal unit*). Sideprotokoll põhineb klient-server arhitektuuril, milles üks liidriseade (ülem, *master*), mida võib nimetada kliendiks teostab mitmete alluvseadmete (*slave*) päringuid. Alluvseadmeid võib siin nimetada ka serveriteks. Liidriseade (ülemseade) saadab käsu vastavale alluvseadmele. Alluvseade töötleb käsu ja saadab vastuse. Kõik sideseansid algatatakse liidriseadme (ülemseadme, *master*) poolt (see on tsentraliseeritud eelisõigusega sidesüsteem).

Temperatuurikontroller XR10CX on võrgus alluvseadmeks (*slave*) ega saada infot võrku enne kui ülemseade (*master*) teeb vastava päringu. Protokoll MODBUS RTU kasutatakse laialdaselt hooneautomaatikas (*Building Management Systems, BMS, Building Automation Systems, BAS*) ja tööstusautomaatikasüsteemides (*Industrial Automation Systems, IAS*). Lai kasutusvaldkond tuleneb protokoll kasutamise lihtsusest erinevatel seadmetel. MODBUS RTU protokoll on madalatasemeline protokoll, milles puuduvad seadmetüübid, andmete skaleerimine ja andmete kirjeldamine. Seadistatakse vaid nimekiri aadressidest millelt saab lugeda (teostada andmete päringuid) ja millesse saab kirjutada (väärtusi saata). Vastuvõtjal peab olema seadistatud andmetüübid, selleks et edastatavaid andmeid oleks võimalik õigesti kasutada. Andmetüübid ja registrid määratakse enamasti sama nimekirjaga, mis on sageli antud kasutatava seadme dokumentatsioonis.

#### Sideprotokoll detailne kirjeldus

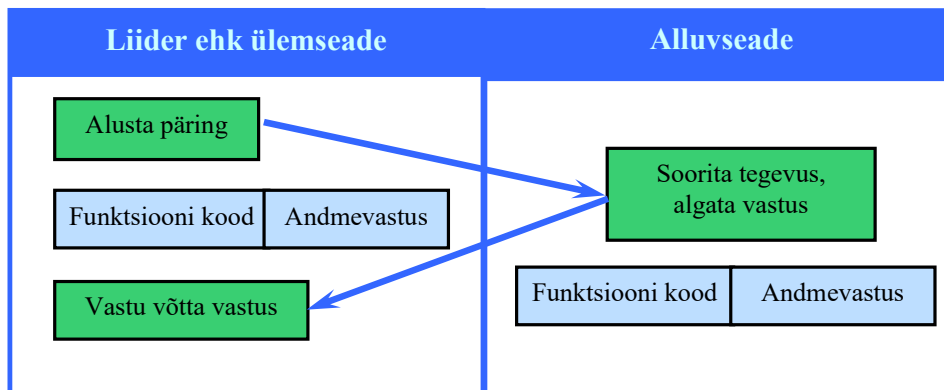
MODBUS-protokoll rakenduse andmeplokk pannakse kokku liidriseadme (ülem, *master*) ehk päringuid teostava kliendi poolt. Edastatav funktsioon näitab alluvseadmele (serverile), millist tegevust teha. Funktsioon on MODBUS-protokoll andmekaadris (andmepakettides) kodeeritud (ette antud) ühe baitiga.

Kui sõnum saadetakse liidriseadmelt (*master*) alluvseadmele (*slave*), siis funktsioonikood väljaga määratakse alluva poolt teostatav tegevus. Ülejäänud teade sisaldab etteantud tegevuse teostamiseks vajalikku informatsiooni. Kui sõnum võetakse vastu vigadeta, siis vastab alluvseade liidrile omapoolse andmepaketiga (vastusega). Vastuse sõnum sisaldab teavet, mida alluv kogus etteantud funktsiooni koodiga määratud toimingute tegemisel. Kui etteantud funktsioonis andmete edastust polnud ettenähtud, siis kordab alluv (*slave*) liidriseadmelt (*master*) saadud käsku. Eduka andmevahetuse skeem on joonisel 1.

Alluvseade ei suuda käsku täita, kui võeti vastu näiteks defineerimata ja vigane käsk või kättesaamatu aadress, siis sisaldab tema vastuse sõnum veakoodi ja

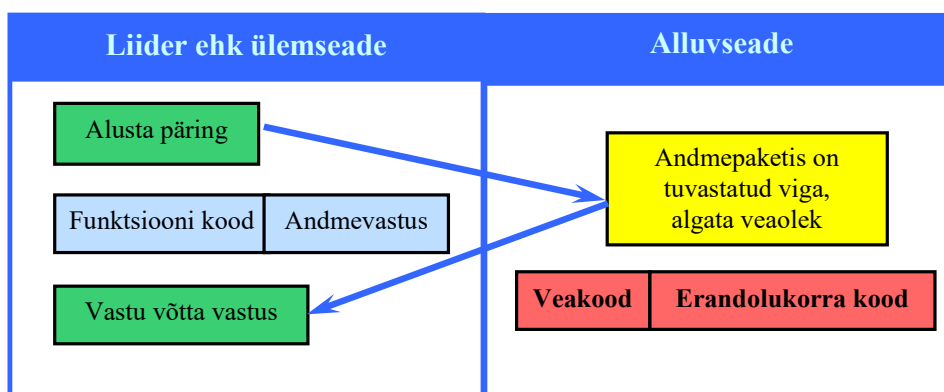
erandolukorra koodi (*exception code*). Erandvastusega andmevahetuse skeem on joonisel 2. Erandolukorra koodide (*exception codes*) loetelu leiab tabelist 1.

### Andmevahetus tavaolukorras



**Joonis 1.** MODBUS-protokolli andmevahetus ilma vigade esinemiseta

### Andmevahetus erandolukorra olekukoodiga



**Joonis 2.** MODBUS-protokolli andmevahetus veakoodi (*error*) ja erandolukorra olekukoodiga (*exception code*)

### Andmete vahetamise järjestus MODBUS RTU protokolliga kasutamisel:

1. Automaatikasüsteemi liidriseade (*master*) saadab käsu alluvseadmele (nt termostaatkontrollerile XR10CX)
2. Pärast teate edastamist lülitab liidriseade oma liini-tüürahelad (*line drivers*) välja ja alustab info vastuvõtu ootamist liinilt.
3. Kõik alluvseadmed võtavad vastu saadetud käsu (alluvseadmed on tavaolekus info vastuvõtturežiimis)
  - a) Kui vastuvõetud aadress vastab alluvseadme aadressile ja kontrollkood (CRC) on õige, siis alluvseade töötleb vastuvõetud käsu.
    - i) Käsu eduka täitmise korral: (1) Vastab alluvseade oma aadressiga ja saadab tagasi vastuvõetud käsu, lisades sellele andmed juhul kui need olid liidriseadme poolt käsus ettenähtud. (2) Pärast vastuse saatmist jätkab

alluvseade järgnevate käskude vastuvõtu ootamist (*listen mode*).

- ii) Kui käsu töötlemine või täitmine ebaõnnestus, siis: (1) Vastab alluvseade erandolukorra koodiga, mis annab infot käsu mittetäitmise põhjustest. Võimalikus põhjuseks on näiteks vale aadress (vt. erandolukorra koodide tabel 1). (2) Pärast erandolukorra vastuse saatmist läheb alluvseade tagasi järgnevate käskude vastuvõtu ootamisele (*listen mode*).
- b) Kui saadud käsu kontrollkood (CRC) ei ole õige, siis vastust ei saadeta.
- c) Kui saadud käsu aadress ei vasta alluvseadme (slave) aadressile, siis vastust ei saadeta.
- d) Kui käsk on poolik, siis vastust ei saadeta.
- e) Kui käsku vastu ei võetud, siis vastust ei saadeta.

## 1. Termostaatkontrolleri DIXELL XR10CX käsuteated

Protokolli MODBUS standardis on ettenähtud mitmeid erinevaid käske erinevat liiki andmetele juurdepääsuks (päringuks või edastamiseks). Termostaadil XR10CX on kasutusel andmetüübid “hoiuregister” (*holding register*), mille aadressid on vahemikus 40001 – 49999 ja väljud ehk “väljundmähis” (*output coil*), mille aadressid on vahemikus 00001 – 09999. Hoiuregistritele juurdepääsuks on vaja ainult kolme käsku ja väljunditele (mähistele) ligipääsuks kahte käsku. Protokoll MODBUS võimaldab lugeda või kirjutada (salvestada) mitmeid järjestikuliste aadressidega registreid kuid ühe käsuga saab korraga kirjutada (salvestada) või lugeda kuni 5 registri väärtust. Releede mähiste olekuid saab korraga järjest lugeda kuni 80 ja kirjutada (muuta) saab termostaadil XR10CX korraga ainult 1 rele mähise olekut.

Järgnevates käskudes kasutatakse tegelikku aadressi ja käsukood määrab registritüübi, mille jaoks vastavat käsku kasutatakse. Näiteks esimesel registril 40001 on aadressiks 0 ja esimese releemähisel 00001 on samuti aadressiks 0.

### 1.1. Väljund releede mähiste oleku lugemine käsuga 0x01=READ OUTPUT COIL

Käsuga 0x01 toimub väljundreleede mähiste olekute lugemine seadmest. Mähiste olekud (1 või 0) pakitakse vastuse sõnumi andmebaitidesse. Baitides on kasutusel ja vastab tegelikkusele info ainult päringus sisaldunud mähiste oleku kohta, kõigi täiendavate st paketi täiteks lisatud bittide ja baitide väärtusi tuleb ignoreerida. Liiderseade (ülemseade) saadab alluvseadmetele mähiste oleku lugemiseks andmepaketi järgnevas vormingus.

Alluv-seadme aadress	Funkt-siooni kood 0x01	Mähise aadressi suurima kaaluga bait (MSByte)	Mähise aadressi vähima kaaluga bait (LSByte)	Mähiste arvu suurima kaaluga bait (MSByte)	Mähiste arvu vähima kaaluga bait (LSByte)	Kontroll koodi CRC vähima kaaluga bait (LSByte)	Kontroll kood CRC suurima kaaluga bait (MSByte)

**Alluvseadme aadress:** (1 bait): Seadme aadress, mis käskusid vastu võtab. Seadmete aadressid on vahemikus 1-247.

**Funktsiooni (käsu) kood:** (1 bait): Koodile 0x01 vastab väljundi mähise oleku

lugemise käsk (*Read Output Coil*).

**Relee mähise aadress:** (2 baiti): Esimese relee mähise aadress, mille olekut loetakse. Mitme relee mähise oleku lugemine toimub järjekorras.

**Releede mähiste arv:** (2 baiti): Elementide (näiteks releede mähiste) arv, mille kohta peab seade infot tagastama (nt. arvutle 3 vastab 3 mähist või 3 bitti infot). Võimalik on teha korraga päringut maksimaalselt kuni 80 elemendi (nt relee mähise) kohta, millest igäühe kohta on 1 bitt infot.

**Kontrollkood (CRC):** (2 baiti): Kontrollkood CRC arvutatakse iga vastuvõetud andmepaketi kohta ja seda kasutatakse kontrollimaks vastuvõetud andmepaketi terviklikkust (*data integrity*).

### Vastusevormingud

Käsu eduka täitmise korral laekub alluvseadmelt liidriseadmele (ülemseadmele) päringu vastusena tagasi pakett järgnevas vormingus:

Alluv-seadme aadress	Käsu eduka täitmise kinnitus kood 0x01	Baitide arv paketis	Andmed 1 (bait)	Andmed ... (baidid)	Andmed n (bait)	Kontrollkoodi CRC vähima kaaluga bait (LSByte)	Kontrollkoodi CRC suurima kaaluga bait (MSByte)
----------------------	--	---------------------	-----------------	---------------------	-----------------	--	---

**Alluvseadme aadress:** (1 bait): Alluvseade tagastab vastuseteates enda aadressi st sama aadressi mis oli andmevahetuse algatanud käsus.

**Funktsiooni (käsu) kood:** (1 bait): Käsu eduka täitmise korral tagastab alluvseade oma vastuses täidetud käsu koodi (antud juhul 0x01).

**Baitide arv paketis:** (1 bait): Määrab edastatavas teates järgnevate baitide arvu arvestamata kontrollkoodi (CRC) baite.

**Andmed [1..n]:** Edastatavate andmete hulk (puhver), mille pikkus baitides oli eelnevalt määratud baitide arvuga paketis. Kirjeldatud käsu (0x01) korral on selles releemähiste olekud st esimesed 8 mähise olekut on salvestatud esimeses baidis.

**Kontrollkood (CRC):** (2 baiti):

Käsu täitmise ebaõnnestumisel st vea tekkimise korral edastatakse andmepaketi asemel erandolukorra vastus (*exception response*), mille vorming on järgmine:

Alluvseadme aadress	Erandolukorra vastuse tunnus (Exception Response) 0x01 + 0x80 Veakood: 0x81	Erandolukorra kood (Exception Code) vt. erandi koodide nimekirja	Kontrollkoodi CRC vähima kaaluga bait (LSByte)	Kontrollkoodi CRC suurima kaaluga bait (MSByte)
---------------------	--	--	--	---

**Alluvseadme aadress:** (1 bait): Alluvseadme aadress, mis tagastab erandolukorra vastuse. See on sama aadress mille eelnevalt edastas liidriseade (ülemseade) alluvseadmele andmevahetust algatanud käsus.

**Erandolukorra vastuse tunnus (Exception Response):** (1 bait): Erandolukorra vastuse

tunnus (nt vea tunnus) saadakse käsukoodile baidi suurima kaaluga biti liitmisega. Näiteks käsukoodi 0x01 korral tagastatakse vea korral  $0x01+0x80 = 0x81$ .

**Erandolukorra kood (Exception Code):** (1 bait): Erandolukorra koodide sh veakoodide selgitusi vaata tabelist 1.

**Kontrollkood (CRC):** (2 baiti):

## 1.2. Väljundi relee mähise oleku kirjutamine käsuga 0x05=WRITE SINGLE COIL

Väljundi relee saatmisel (kirjutamisel) käsuga 0x05 (WRITE SINGLE COIL) saab termostaadil XR10CX korraga muuta ühe relee mähise olekut (mähist sisse või välja lülitada). Mitme väljundirelee oleku edastamiseks tuleb kasutada mitut käsku. Mähise olek seatakse (SET, 1) saates käsu andmeväljal koodi 0xFF00. Mähise olek nullitakse (RESET, 0) saates käsu andmeväljal koodi 0x0000.

Alluv-seadme aadress	Funktsiooni kood 0x05	Mähise aadressi suurima kaaluga bait (MSByte)	Mähise aadressi vähima kaaluga bait (LSByte)	Andmete suurima kaaluga bait (MSByte)	Andmete vähima kaaluga bait (LSByte)	Kontrollkoodi CRC vähima kaaluga bait (LSByte)	Kontrollkoodi CRC suurima kaaluga bait (MSByte)
-------------------------	-----------------------------	---	--	---	--	---	--

**Alluvseadme aadress:** (1 bait): Seadme aadress, mis käskusid vastu võtab. Seadmete aadressid on vahemikus 1-247.

**Funktsiooni (käsu) kood:** (1 bait): Koodile 0x05 vastab väljundirelee mähise oleku kirjutamise käsk (*Write Output Coil*).

**Mähise aadress:** (2 baiti): Mähise aadress, mille väärtust saadetakse.

**Andmed:** (2 baiti): Kood 0x000 nullib (RESET, 0) mähise oleku ja kood 0xFF00 seab mähise oleku üheks (SET, 1).

**Kontrollkood (CRC):** (2 baiti): Kontrollkood CRC arvutatakse iga vastuvõetud andmepaketi kohta ja seda kasutatakse kontrollimaks vastuvõetud andmepaketi terviklikkust (*data integrity*).

## Vastusevormingud

Käsu eduka täitmise korral laekub alluvseadmelt liidarseadmele (ülemseadmele) päringu vastusena tagasi pakett järgnevas vormingus:

Alluv-seadme aadress	Käsu eduka täitmise kinnitus kood 0x05	Mähise aadressi suurima kaaluga bait (MSByte)	Mähise aadressi vähima kaaluga bait (LSByte)	Andmete suurima kaaluga bait (MSByte)	Andmete suurima kaaluga bait (LSByte)	Kontroll- koodi CRC vähima kaaluga bait (LSByte)	Kontroll- koodi CRC suurima kaaluga bait (MSByte)
-------------------------	--	--	---	---	---	---	--

**Alluvseadme aadress:** (1 bait): Alluvseade tagastab vastuseteates enda aadressi st sama aadressi mis oli andmevahetuse algatanud käsus.

**Funktsiooni (käsu) kood:** (1 bait): Käsu eduka täitmise korral tagastab alluvseade oma vastuses täidetud käsu koodi (antud juhul 0x05).

**Mähise aadress:** (2 baiti): Saadetakse tagasi (*echo*) täidetud käsus sisaldunud mähise aadress.

**Andmed:** (2 baiti): Saadetakse tagasi (*echo*) täidetud käsus sisaldunud andmed.

**Kontrollkood (CRC):** (2 baiti):

Käsu täitmise ebaõnnestumisel st vea tekkimise korral edastatakse andmepaketi asemel erandolukorra vastus (*exception response*), mille vorming on järgmine:

Alluvseadme aadress	Erandolukorra vastuse tunnus ( <i>Exception Response</i> ) 0x05 + 0x80 Veakood: 0x85	Erandolukorra kood ( <i>Exception Code</i> ) vt. erandi koodide nimekirja	Kontrollkoodi CRC vähima kaaluga bait (LSByte)	Kontrollkoodi CRC suurima kaaluga bait (MSByte)
---------------------	---	---	--	---

**Alluvseadme aadress:** (1 bait): Alluvseade tagastab vastuseteates enda aadressi st sama aadressi mis oli andmevahetuse algatanud käsus.

**Erandolukorra vastuse tunnus (*Exception Response*):** (1 bait): Erandolukorra vastuse tunnus (nt veatunnus) saadakse käsekoodile baidi suurima kaaluga biti liitmisega. Näiteks käsekoodi 0x05 korral tagastatakse vea korral  $0x05+0x80 = 0x85$ .

**Erandolukorra kood (*Exception Code*):** (1 bait): Erandolukorra koodide sh veakoodide selgitusi vaata tabelist 1.

**Kontrollkood (CRC):** (2 baiti):

### 1.3. Seadme hoiuregistrite lugemine käsuga 0x03=READ HOLDING REGISTERS

Käsuga 0x03 (READ HOLDING REGISTERS) toimub registrites sisalduvate väärtuste lugemine seadmest. Käsu eduka täitmise korral sisaldab vastusepakett seadme registrites sisaldunud väärtusi. Liiderseade (ülemseade) saadab alluvseadmetele mähiste oleku lugemiseks andmepaketi järgnevas vormingus.

Alluvseadme aadress	Funkt-siooni kood 0x03	Registri aadressi suurima kaaluga bait (MSByte)	Registri aadressi vähima kaaluga bait (LSByte)	Registrite arvu suurima kaaluga bait (MSByte)	Registrite arvu vähima kaaluga bait (LSByte)	Kontrollkoodi CRC vähima kaaluga bait (LSByte)	Kontrollkoodi CRC suurima kaaluga bait (MSByte)
---------------------	---------------------------	---	--	---	--	--	---

**Alluvseadme aadress:** (1 bait): Seadme aadress, mis käskusid vastu võtab. Seadmete aadressid on vahemikus 1-247.

**Funkt-siooni (käsu) kood:** (1 bait): Koodile 0x03 vastab hoiuregistrite väärtuste lugemise käsk (Read holding register).

**Registri aadress:** (2 baiti): Esimese registri aadress, mille väärtust loetakse. Mitme registri väärtuse lugemine toimub järjekorras.

**Registrite arv:** (2 baiti): Elementide (näiteks registrite) arv, mille kohta peab seade infot tagastama (nt. arvutle 3 vastab 3 registrit ehk 3x16 bitti infot). Võimalik on teha korraga päringut maksimaalselt kuni 5 registri kohta, millest igaüks sisaldab 16 bitti

infot.

**Kontrollkood (CRC):** (2 baiti): Kontrollkood CRC arvutatakse iga vastuvõetud andmepaketi kohta ja seda kasutatakse kontrollimaks vastuvõetud andmepaketi terviklikkust (*data integrity*).

### Vastusevormingud

Käsu eduka täitmise korral laekub alluvseadmelt liidarseadmele (ülemseadmele) päringu vastusena tagasi pakett järgnevas vormingus:

Alluvseadme aadress	Käsu eduka täitmise kinnitus kood 0x03	Baitide arv pakettis	Andmed 1 (bait)	Andmed ... (baidid)	Andmed n (bait)	Kontrollkoodi CRC vähima kaaluga bait (LSByte)	Kontrollkoodi CRC suurima kaaluga bait (MSByte)
---------------------	--	----------------------	-----------------	---------------------	-----------------	--	---

**Alluvseadme aadress:** (1 bait): Alluvseade tagastab vastuseteates enda aadressi st sama aadressi mis oli andmevahetuse algatanud käsus.

**Funktsiooni (käsu) kood:** (1 bait): Käsu eduka täitmise korral tagastab alluvseade oma vastuses täidetud käsu koodi (antud juhul 0x03).

**Baitide arv pakettis:** (1 bait): Määrab edastatavas teates järgnevate baitide arvu arvestamata kontrollkoodi (CRC) baite.

**Andmed [1..n]:** Edastatavate andmete hulk (puhver), mille pikkus baitides oli eelnevalt määratud baitide arvuga pakettis. Kirjeldatud käsu (0x01) korral on selles releemähiste olekud st esimesed 8 mähise olekut on salvestatud esimeses baidis.

**Kontrollkood (CRC):** (2 baiti):

Käsu täitmise ebaõnnestumisel st vea tekkimise korral edastatakse andmepaketi asemel erandolukorra vastus (*exception response*), mille vorming on järgmine:

Alluvseadme aadress	Erandolukorra vastuse tunnus ( <i>Exception Response</i> ) 0x03 + 0x80 Veakood: 0x83	Erandolukorra kood ( <i>Exception Code</i> ) vt. erandi koodide nimekirja	Kontrollkoodi CRC vähima kaaluga bait (LSByte)	Kontrollkoodi CRC suurima kaaluga bait (MSByte)
---------------------	---	---	--	---

**Alluvseadme aadress:** (1 bait): Alluvseadme aadress, mis tagastab erandolukorra vastuse. See on sama aadress mille eelnevalt edastas liidarseade (ülemseade) alluvseadmele andmevahetust algatanud käsus.

**Erandolukorra vastuse tunnus (*Exception Response*):** (1 bait): Erandolukorra vastuse tunnus (nt veatunnus) saadakse käsukoodile baidi suurima kaaluga biti liitmisega. Näiteks käsukoodi 0x03 korral tagastatakse vea korral  $0x03+0x80 = 0x83$ .

**Erandolukorra kood (*Exception Code*):** (1 bait): Erandolukorra koodide sh veakoodide selgitusi vaata tabelist 1.

**Kontrollkood (CRC):** (2 baiti):



#### 1.4. Seadme registri kirjutamine käsuga 0x06=WRITE SINGLE REGISTER

Käsuga 0x06 (WRITE SINGLE REGISTER) toimub väärtuse kirjutamine seadme registrisse. Käsu eduka täitmise korral sisaldab vastusepakett seadme registrisse kirjutatud väärtust. Liidriseade (ülemseade) saadab alluvseadmetele registri väärtuse kirjutamiseks andmepaketi järgnevas vormingus.

Alluv-seadme aadress	Funktsiooni kood 0x06	Registri aadressi suurima kaaluga bait (MSByte)	Registri aadressi vähima kaaluga bait (LSByte)	Andmete suurima kaaluga bait (MSByte)	Andmete vähima kaaluga bait (LSByte)	Kontrollkoodi CRC vähima kaaluga bait (LSByte)	Kontrollkoodi CRC suurima kaaluga bait (MSByte)
----------------------	-----------------------	---	--	---------------------------------------	--------------------------------------	--	---

**Alluvseadme aadress:** (1 bait): Seadme aadress, mis käskusid vastu võtab. Seadmete aadressid on vahemikus 1-247.

**Funktsiooni (käsu) kood:** (1 bait): Koodile 0x06 vastab registri väärtuse kirjutamise käsk (*Write single register*).

**Registri aadress:** (2 baiti): Registri aadress, mille väärtust saadetakse.

**Andmed:** (2 baiti): Registrisse salvestatavad andmed.

**Kontrollkood (CRC):** (2 baiti): Kontrollkood CRC arvutatakse iga vastuvõetud andmepaketi kohta ja seda kasutatakse kontrollimaks vastuvõetud andmepaketi terviklikkust (*data integrity*).

#### Vastusevormingud

Käsu eduka täitmise korral laekub alluvseadmelt liidriseadmele (ülemseadmele) päringu vastusena tagasi pakett järgnevas vormingus:

Alluv-seadme aadress	Käsu eduka täitmise kinnitus kood 0x06	Registri aadressi suurima kaaluga bait (MSByte)	Registri aadressi vähima kaaluga bait (LSByte)	Andmete suurima kaaluga bait (MSByte)	Andmete suurima kaaluga bait (LSByte)	Kontrollkoodi CRC vähima kaaluga bait (LSByte)	Kontrollkoodi CRC suurima kaaluga bait (MSByte)
----------------------	--	---	--	---------------------------------------	---------------------------------------	--	---

**Alluvseadme aadress:** (1 bait): Alluvseade tagastab vastuseteates enda aadressi st sama aadressi mis oli andmevahetuse algatanud käsus.

**Funktsiooni (käsu) kood:** (1 bait): Käsu eduka täitmise korral tagastab alluvseade oma vastuses täidetud käsu koodi (antud juhul 0x06).

**Registri aadress:** (2 baiti): Saadetakse tagasi registri aadress mille väärtus edukalt kirjutati.

**Andmed:** (2 baiti): Andmepuhvri sisu, mis sisaldab sama informatsiooni mida eelnevalt saadeti andmevahetust algatanud käsu andmetena.

**Kontrollkood (CRC):** (2 baiti):

Käsu täitmise ebaõnnestumisel st vea tekkimise korral edastatakse andmepaketi asemel erandolukorra vastus (*exception response*), mille vorming on järgmine:



Alluvseadme aadress	Erandolukorra vastuse tunnus ( <i>Exception Response</i> ) 0x06 + 0x80 Veakood: 0x86	Erandolukorra kood ( <i>Exception Code</i> ) vt. erandi koodide nimekirja	Kontrollkoodi CRC vähima kaaluga bait (LSByte)	Kontrollkoodi CRC suurima kaaluga bait (MSByte)
---------------------	---	---	--	---

**Alluvseadme aadress:** (1 bait): Alluvseade tagastab vastuseteates enda aadressi st sama aadressi, mis sisaldus andmevahetuse algatanud käsus.

**Erandolukorra vastuse tunnus:** (1 bait): Erandolukorra vastuse tunnus (nt veatunnus) saadakse käsukoodile baidi suurima kaaluga biti liitmisega. Näiteks käsukoodi 0x05 korral tagastatakse vea korral  $0x06+0x80 = 0x86$ .

**Erandolukorra kood (Exception Code):** (1 bait): Erandolukorra koodide sh veakoodide selgitusi vaata tabelist 1.

**Kontrollkood (CRC):** (2 baiti):

### 1.5. Seadme hoiuregistrisse kirjutamine käsuga 0x10=WRITE HOLDING REGISTERS

Käsuga 0x10 (WRITE HOLDING REGISTERS) toimub väärtuste paketi kirjutamine ühte kuni viide registrisse. Käsu eduka täitmise korral sisaldab vastusepakett kirjutatud registrite arvu. Liidriseade (ülemseade) saadab alluvseadmetele registrite väärtuste kirjutamiseks andmepaketi järgnevas vormingus.

Alluvseadme aadress	Funktsiooni kood 0x10	Registri aadressi suurima kaaluga bait (MSByte)	Registri aadressi vähima kaaluga bait (LSByte)	Registrite arvu suurima kaaluga bait (MSByte)	Registrite arvu vähima kaaluga bait (LSByte)	Baitide arv paketis	Andmed	Kontrollkoodi CRC vähima kaaluga bait (LSByte)	Kontrollkoodi CRC suurima kaaluga bait (MSByte)
---------------------	-----------------------	---	--	---	--	---------------------	--------	--	---

**Alluvseadme aadress:** (1 bait): Seadme aadress, mis käskusid vastu võtab. Seadmete aadressid on vahemikus 1-247.

**Funktsiooni (käsu) kood:** (1 bait): Koodile 0x10 vastab hoiuregistrisse (1 kuni 5) väärtuse kirjutamise käsk (Write holding registers).

**Registri aadress:** (2 baiti): Esimese registri aadress, mille väärtust kirjutatakse.

**Registrite arv:** (2 baiti): Määrab elementide (näiteks registrite) arvu, millesse toimub informatsiooni salvestamine. Termostaadil XR10CX on ühe käsuga võimalik salvestada infot kuni viide registrisse.

**Baitide arv paketis:** (1 bait): Määrab edastatavas teates järgnevate baitide arvu arvestamata kontrollkoodi (CRC) baite. Baitide arv peab olema kaks korda suurem kui adresseeritavate registrite arv, sest iga 16-bitine register säilitab kahte baiti.

**Andmed:** Registritesse kirjutatavad andmed salvestatakse edastatavasse paketti järjekorras, milles on enne suurema kaaluga bait ja seejärel väiksema kaaluga bait (MSByte, LSByte order).

**Kontrollkood (CRC):** (2 baiti): Kontrollkood CRC arvutatakse iga vastuvõetud andmepaketi kohta ja seda kasutatakse kontrollimaks vastuvõetud andmepaketi terviklikkust (*data integrity*).

## Vastusevormingud

Käsu eduka täitmise korral laekub alluvseadmelt liidriseadmele (ülemseadmele) päringu vastusena tagasi pakett järgnevas vormingus:

Alluvseadme aadress	Käsu eduka täitmise kinnitus kood 0x10	Registri aadressi suurima kaaluga bait (MSByte)	Registri aadressi vähima kaaluga bait (LSByte)	Registrite arvu suurima kaaluga bait (MSByte)	Registrite arvu vähima kaaluga bait (LSByte)	Kontrollkoodi CRC vähima kaaluga bait (LSByte)	Kontrollkoodi CRC suurima kaaluga bait (MSByte)
---------------------	--	---	--	---	--	--	---

**Alluvseadme aadress:** (1 bait): Alluvseade tagastab vastuseteates enda aadressi st sama aadressi mis oli andmevahetuse algatanud käsus.

**Funktsiooni (käsu) kood:** (1 bait): Käsu eduka täitmise korral tagastab alluvseade oma vastuses täidetud käsu koodi (antud juhul 0x10).

**Registri aadress:** (2 baiti): Registri aadress millest alates väärtused (etteantud arv baite) edukalt kirjutati.

**Registrite arv:** (2 baiti): Kirjutatud registrite arv.

**Kontrollkood (CRC):** (2 baiti).

Käsu täitmise ebaõnnestumisel st vea tekkimise korral edastatakse andmepaketi asemel erandolukorra vastus (*exception response*), mille vorming on järgmine:

Alluvseadme aadress	Erandolukorra vastuse tunnus ( <i>Exception Response</i> ) 0x10 + 0x80 Veakood: 0x90	Erandolukorra kood ( <i>Exception Code</i> ) vt. erandi koodide nimekirja	Kontrollkoodi CRC vähima kaaluga bait (LSByte)	Kontrollkoodi CRC suurima kaaluga bait (MSByte)
---------------------	---	---	--	---

**Alluvseadme aadress:** (1 bait): Alluvseade tagastab vastuseteates enda aadressi st sama aadressi mis oli andmevahetuse algatanud käsus.

**Erandolukorra vastuse tunnus:** (1 bait): Erandolukorra vastuse tunnus (nt veatunnus) saadakse käsukoodile baidi suurima kaaluga biti liitmisega. Näiteks käsukoodi 0x05 korral tagastatakse vea korral  $0x10+0x80 = 0x90$ .

**Erandolukorra kood (*Exception Code*):** (1 bait): Erandolukorra koodide sh veakoodide selgitusi vaata tabelist 1.

**Kontrollkood (CRC):** (2 baiti):

### 1.6. Erandolukorra vastuse koodid

Erandolukord tekib kui andmepakett võetakse alluvseadme poolt korrektselt ja vigadeta vastu aga alluvseade ei saa käsku täita. Selle põhjuseks võib olla vigane või vale aadress, või kirjutuskäsk registrisse, mis võimaldab üksnes andmete lugemist (*read only*). Tabelis 1 on toodud erandolukorra koodide (*Exception Code*) selgitused ja võimalikud põhjused.

**Tabel 1. Erandolukorra koodide näiteid**

Erandolukorra kood	Erandolukorra nimetus	Selgitus
01	Vale funktsioon ( <i>Illegal Function</i> )	Alluvseadme poolt vastu võetud funktsiooni (käsu) kood ei vasta alluvseadme poolt lubatud tegevustele. Termostaatkontroller toetab käskudeks ainult koode 0x03, 0x06 ja 0x10.
02	Vale aadress ( <i>Illegal Data Address</i> )	Päringuga saadud andmeaadress ei ole alluvseadme poolt lubatavas või toetatavas vahemikus. See viga esineb ka siis kui andmetes etteantud viitenumbri ja edastatava paketi pikkuse kombinatsioon ei ole õige.
03	Vale andmeobjekt või väärtus ( <i>Illegal Data Value</i> )	Päringus viidatud aadressiga registrit ei ole olemas. Viga tekib ka siis kui päringus küsitakse termostaatkontrollerilt rohkem kui viit elementi. Viga tekib kui funktsiooni parameetri väärtus on piiridest väljas ( <i>out of range</i> ). Samuti siis kui üritatakse kirjutada registrisse mis võimaldab ainult andmete lugemist ( <i>read only</i> ).
04	Alluvseadme viga ( <i>Slave Device Failure</i> )	Alluvseadmes tekkis taastumatu viga kui seade üritas etteantud käsufunktsiooni täita. Alluvseadmel ei õnnestunud täita oma sisemise mälu või registre kirjutamist või lugemist. Vea võimalikeks põhjusteks on ka muutmälu (RAM), E2 või kella (RTC) ebakorrekne talitlus.
06	Alluvseade hõivatud ( <i>Slave Device Busy</i> )	Seade ei saanud päringu käsufunktsiooniga antud hetkel täita. Võimalikuks põhjuseks on näiteks teise analoog-digitaalmuunduri tsükli töö. Liidriseade (ülemseade) peaks sama päringut mõne aja pärast kordama.

## 2. Termostaatkontrolleri XR10CX registrid ja mähised

### 2.1. Termostaatkontrolleri XR10CX registrid

**Tabel 2** Termostaatkontrolleri XR10CX registre nimekiri.

Tähis (tabeli esimeses tulpas) vastab termostaatkontrolleri näidikul kuvatavale tähisele või klemmide siltidel olevatele tähistele.

Tähis	Suund	Kirjeldus	Mähis	Register
PRB1	Lugemine	Prooviku 1 temperatuur		40257
HY	Lugemine/Kirjutamine	Diferents (erinevus)		40769
ALd	Lugemine/Kirjutamine	Temperatuurialarmi viide		40830
DAo	Lugemine/Kirjutamine	Temperatuurialarmi viide käivitamisel		40831
Did	Lugemine/Kirjutamine	Digitaalsisendi alarmi viide minutites (0-255)		40847
Adr	Lugemine/Kirjutamine	Seadme MODBUS protokolliga aadress 1-247		40852
Set	Lugemine/Kirjutamine	Süsteemi seadesuurus ( <i>System Setpoint</i> )		40864
<b>Coils</b>				

Tähis	Suund	Kirjeldus	Mähis	Register
	Lugemine/Kirjutamine	Sisse-välja lülitamise juhtsignaal	513	
	Lugemine/Kirjutamine	Klahvistiku lukk	516	
	Lugemine/Kirjutamine	Summeri ja kõigi alarmide (Any) vaigistamine	517	
	Lugemine	Digitaalsisendi klemmi 9 olek	520	
	Lugemine	Prooviku 1 viga	521	
	Lugemine	Prooviku 2 viga	522	
	Lugemine	Prooviku 3 viga	523	
	Lugemine	Prooviku 4 viga	524	
	Lugemine	Kõrge temperatuur alarm Pb1	525	
	Lugemine	Madala temperatuuri alarm Pb1	526	
	Lugemine	Viga välise sisendi ( <i>external input</i> ) kaudu	529	
	Lugemine	EEPROM-mälu viga	532	
	Lugemine	Kõiki ( <i>Any</i> ) alarme saab vaigistada	538	
	Lugemine	Relay Call For Heat	543	
	Lugemine	Summer on sisselülitatud välise või kõrgete (Hi) vigade korral	545	

## 2.2. MODBUS-protokolliga hallatavad parameetrid

Termostaatkontroller võimaldab MODBUS protokolliga ligipääsu järgmiste parameetrite väärtusi sisaldavatele registritele:

- Seadesuuruse register 40864**  
 Selle kaudu toimub seadesuuruse etteandmine süsteemile.  
 Register on nii loetav kui ka kirjutatav (Read/Write).
- MODBUS võrgu ID ehk aadressi register 40852**  
 Selles registris säilitatakse seadme MODBUS aadressi, mis on vahemikus 0-247  
 Seda aadressi tuleks seadistada seadme klahvistiku abil.
- Prooviku temperatuuri register 40257**  
 (TP1) Prooviku 1 temperatuur Fahrenheiti kraadides (F) (40257)  
 See on talitlustemperatuuri proovik mida kasutatakse süsteemi temperatuuri etteandmiseks (register on ainult loetav).
- Väljundreele (ahju juhtimise) register 543**  
 See annab ette väljundreele oleku. Kui see signaal on 1 siis on ahju juhtsignaal sisselülitatud.
- Alarmide register (Modbus Coil) 538**  
 Seda tüüpi seadmetel harilikult alarme ei kasutata .
- Juhtimise sisse ja väljalülitamise (Enable/Disable) register (Modbus Coil) 513**  
 ON=257 or 0x0101, OFF=1 or 0x0001  
 Sisselülitatud (lubatud, ON) olekule vastab kood 0x0101 ehk 257. Väljalülitatud (keelatud, OFF) olekule vastab koos 0x0001 ehk 1. Keelatud (OFF) olekus ei

rakenda kontrollid küttesignaale, alarme, kõik kontrolleri releeväljundid on lahutatud olekus ja temperatuure ei kuvata. Sel juhul kuvatakse kontrolleri näidikul teksti „OFF“. Sellegipoolest on võimalik lugeda seadmest proovikute temperatuure ja muid parameetreid MODBUS võrgu kaudu. See on ainult kirjutatav (*write only*) register ja selle lugemine annab alati valesid tulemusi.

### *6.2 Ülesande kirjeldus 1*

Koostage MODBUS andmepakett Dixel-temperatuurikontrolleri päringuks ja vastuse töötlusprogramm digitaalse sisendi oleku lugemiseks. Päringu teostamiseks kasutage funktsiooni READ OUTPUT COIL (0x01).

### *6.3 Ülesande kirjeldus 2*

Koostage MODBUS andmepakett jahutusseadme sisse- või väljalülitamiseks Dixeli temperatuurikontrolleri kaudu ja selle vastuse töötlusprogramm. Kasutage funktsiooni WRITE SINGLE COIL (0x05).

### *6.4 Ülesande kirjeldus 3*

Koostage MODBUS päringu pakett jahutusseadme temperatuuri hetkeväärtuse päringuks Dixel-temperatuurikontrollerist ja programm vastuse töötlemiseks. Päringu teostamiseks kasutage funktsiooni READ HOLDING REGISTERS (0x03).

### *6.5 Aruande struktuur*

Aruande pikkus ei tohiks ületada kolme lehekülge. Aruanne laadida e-õppe platvormile (Workspaces).

### *6.6 Kirjandusviited*

1. MODBUS-RTU Applied to the XR10CX Control: Water heater digital operating controller, PVI Industries, LLC, PV500-67 06/2014  
<https://www.ccontrols.com/support/dp/PV500-67.pdf>