



Tööjuhend

Terraariumi seirelahendus: riistvara, pilvepool ja näidisprogrammid

Autorid: Olle Arak, Agnes Heinla

Tööjuhend on koostatud Saaremaa Gümnaasiumi projekti „Targad rohelised lahendused“ raames.

Saaremaa Gümnaasium 2026

Juhend kirjeldab terraariumi seirelahendust – seadet, pilvepoolt ja nende töös hoidmist – inimesele, kes seadmed kokku paneb ja neid haldab (õpetaja, IT-juht, huviringi juhendaja), sõltumata varasemast programmeerimiskogemusest. Terraariumi anuma ettevalmistus, andurite paigutus anumas, taimed ja korgi sulgemine on eraldi tööjuhendis „Klaasterraariumi kokkupanek komplektist ja seireseadme paigaldamine“. Juhend annab valmis lahenduse: plaadi programmi ja pilvetabeli veebirakenduse, milles tuleb muuta kokku viis väärtust. Sama lahendus töötab nii ühe kui ka kolmekümne seadmega.

Kuidas juhendit ja näidiseid kasutada

Töö käib peatükkide järjekorras ja iga peatükk lõpeb kontrolliga, mis näitab, kas võib edasi minna:

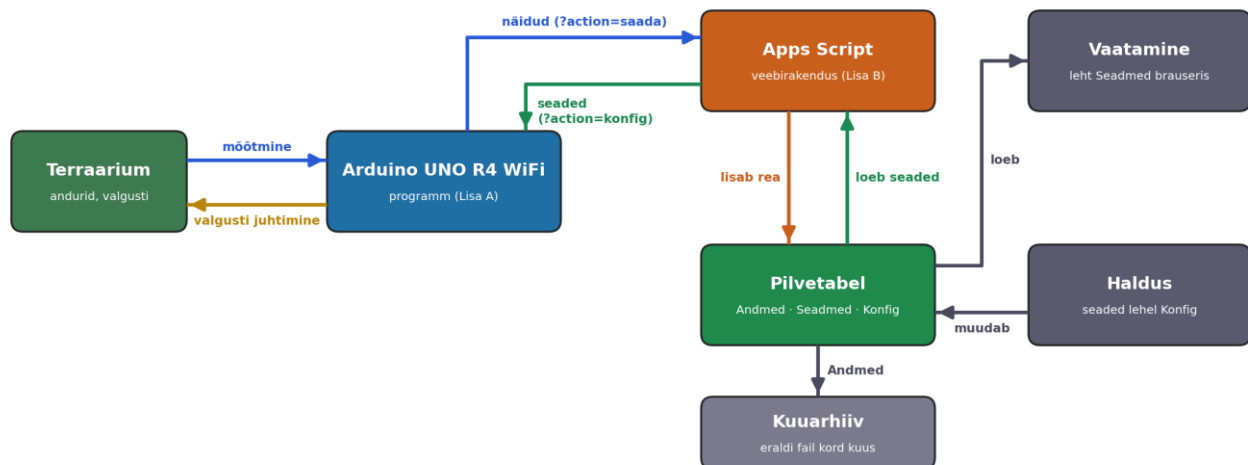
Peatükk	Tulemus	Kontroll
1 Süsteemi ülevaade	arusaam, mis kus töötab	—
2 Riistvara, 3 Kokkupanek	seade laual ühendatuna, toیتeta	ühendustabel läbi vaadatud; iga rea on üle kontrollinud kaks inimest
4 Pilvepool	tabel ja veebirakendus töötavad brauserist	sammud 6–7: rida tekib, seaded tulevad vastusena
5 Plaadi programm	seade saadab andmeid	samm 7: rida tabelis seadme ID-ga
6 Seadistamine ja kalibreerimine	valgusti ja mulla protsent õiged	kontrollnimekiri peatükis 7
7 Mitme seadme tootmine	kõik seadmed tabelis	iga seadme kontrollnimekiri täidetud
8 Hooldus ja vead	seadmed püsivad töös	leht Seadmed iga päev
9 Keelemudel	muudatused ja vead ilma programmeerijata	—

Näidisprogramme on kaks: **Lisa A** on plaadi programm (Terraarium_naidis.ino) ja **Lisa B** on pilvetabeli veebirakendus (Kood_naidis.gs). Mõlemad on juhendi lõpus täies pikkuses. Lisa tekst kopeeritakse tervikuna, esimesest reast viimaseni, ja kleebitakse sihtkohta: Lisa B Apps Scripti redaktorisse (peatükk 4, samm 2), Lisa A Arduino arenduskeskkonda (peatükk 5, samm 4). Kood kopeeritakse Wordi failist; PDF-failist kopeerides satuvad koodi vahele lehekülgede jaluse read, mis tuleb enne kleepimist eemaldada.

Näidistes muudetakse ainult viit väärtust, kõik programmi alguses: Lisa B real TABEL_ID pilvetabeli identifikaator (ID) ning Lisa A ridadel SSID (WiFi-võrgu nimi, ingl Service Set Identifier), PAROOL, SEADE_ID ja WEBAPP_PATH. Kõik muud seaded (valgusti lülituspiirid, heledus, mulla kalibreerimine, saatmisvahe) on pilvetabeli lehel Konfig ja neid programmi ei kirjutata. Kui näidist on vaja muuta, on selleks peatükk 9.

1 Süsteemi ülevaade

Süsteemi ülevaade: andmed lähevad pilve, seaded tulevad pilvest



Joonis 1. Andmevoog: seade saadab näidud pilve ja loeb sealt oma seaded

Seirelahendus koosneb asjade interneti (ingl Internet of Things, IoT) seadmetest ja pilvepoolest: seadmel töötab programm Lisa A ja pilves veebirakendus Lisa B. Iga seade on Arduino UNO R4 WiFi plaat nelja lisamooduliga: fotoresistori moodul (valgus), mahtuvuslik mullaniiskusandur, DHT22 (õhu temperatuur ja suhteline õhuniiskus) ja väljatransistoriga lülitusmoodul IRF520 (MOSFET-moodul, ingl metal–oxide–semiconductor field-effect transistor), mis lülitab eeltakisti kaudu 3 V valgusdiodvalgustit (LED-valgustit, ingl light-emitting diode).

Plaat loeb andureid iga 10 s järel ja saadab iga saada_min minuti järel (vaikimisi 5 min) näidud turvalise hüpertexti edastusprotokolli päringuga (ingl HyperText Transfer Protocol Secure, HTTPS) Google Apps Scripti veebirakendusele. Veebirakendus lisab rea pilvetabeli lehele Andmed, uuendab lehel Seadmed seadme viimast olekut ja vastab plaadile, kas rida salvestati. Iga konfig_min minuti järel küsib plaat lehelt Konfig oma seaded, mille veebirakendus saadab struktureeritud tekstivormingus JSON (ingl JavaScript Object Notation), ja salvestab need püsिमällu (ingl Electrically Erasable Programmable Read-Only Memory, EEPROM), et pärast volukatkestust töötada õigete seadetega ka ilma võrguta. Kord kuus kopeerib veebirakendus lehe Andmed eraldi arhiivifaili (peatükk 7).

Automaatselt juhitakse ainult valgustit. Kastmine ja õhutamine jäävad inimese teha: lahendus ei saada teavitusi, vaid näitab pilvetabeli lehel Seadmed staatust, mida vaadatakse regulaarselt.

Lahendus eeldab Google'i kontot, millel pilvetabel ja veebirakendus asuvad, ning 2,4 GHz WiFi-võrku, mis laseb seadme otse internetti (ilma sisselogimislehta ehk ingl captive portal). Kontoks sobib kooli ühiskonto: veebirakendus töötab konto omaniku nimel ja lakkab töötamast, kui konto suletakse. Google Workspace'i konto puhul peab administraator lubama veebirakenduse avaldamise kõigile (valik Anyone, peatükk 4, samm 5), sest plaat ei logi sisse. Kooli külalisvõrk sobib, kui see eelnimetatud tingimusi täidab. Kõik plaadid saadavad samale aadressile ja erinevad üksteisest ainult seadme ID poolest.

2 Riistvara ühe seadme kohta

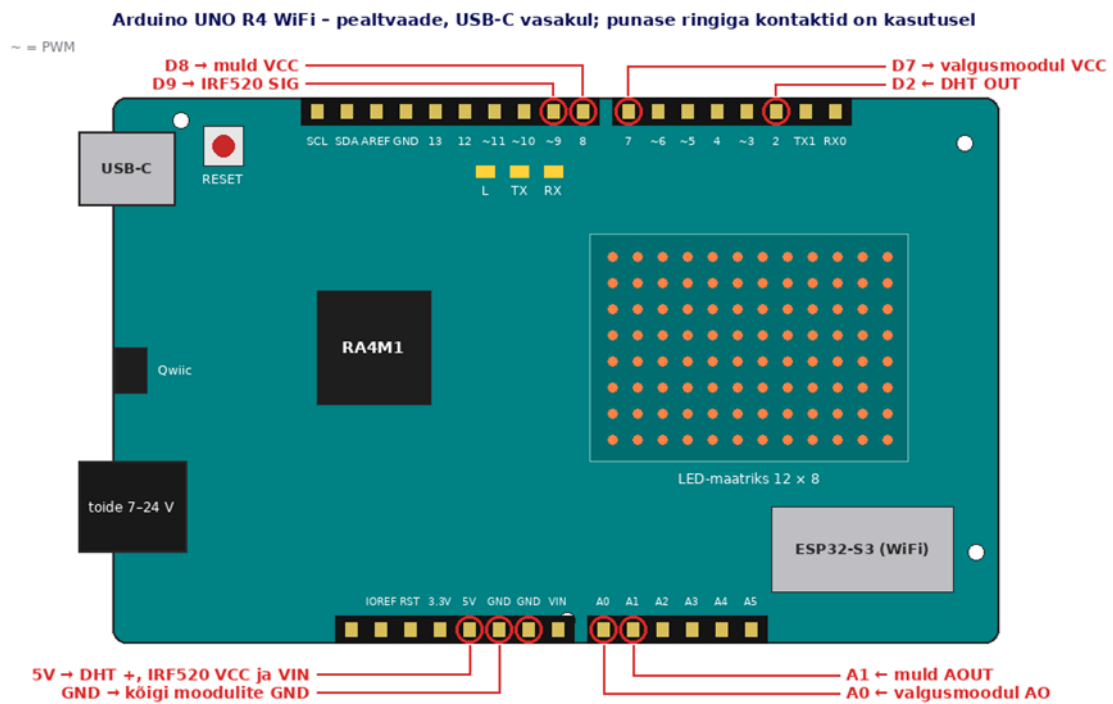
Komponent	Kogus	Märkus
Arduino UNO R4 WiFi + USB-C andmekaabel	1	plaadipakett „Arduino UNO R4 Boards“ arenduskeskkonnas; USB – universaalne jadasiin (ingl Universal Serial Bus)
Fotoresistori moodul, 4 kontakti (VCC, GND, DO, AO)	1	kasutatakse ainult analoogväljundit AO; DO ja potentsiomeeter jäävad kasutamata
Mahtuvuslik mullaniiskusandur v1.2 (VCC, GND, AOUT)	1	mulda ja kalibreerimisel vette täpselt trükitud jooneni; kontaktide järjekord võib plaaditi erineda, ühendatakse trükitud tähiste järgi
DHT22-moodul, 3 kontakti	1	tõmbetakisti on moodulil; mõõtepiirkonna ülaosas (üle 80 %) on mõõteviga suurem, vt peatükk 6
MOSFET-moodul IRF520 (SIG, VCC, GND; VIN/GND; V+/V-)	1	töötab 5 V juhtsignaaliga; 3,3 V plaatidega mitte
LED-valgusti 3 V	1	kinnitatakse korgi alumisele küljele
Eeltakisti valgustile, $\geq 0,25$ W	1	väärtus arvutatakse valgusti voolu järgi, vt allpool
Maketeerimisplaat või joodetud prototüüpplaat, ühendusjuhtmed	1 + ~15	püsipaigalduseks jootmine, vt peatükk 3
Niiskuskindel karp plaadile ja moodulitele	1	karp seisab anuma kõrval
USB-toiteplokk 5 V ≥ 1 A ja USB-C kaabel	1	lisatoide (aku, päikesepaneel) on eraldi dokumendis

Moodulite kontaktide tähised: VCC – toitepinge, GND – maandus (0 V), AO ja AOUT – analoogväljund, DO – digitaalväljund, SIG – juhtsignaal, VIN – koormuse toite sisend, V+ ja V- – koormuse (valgusti) ühendus.

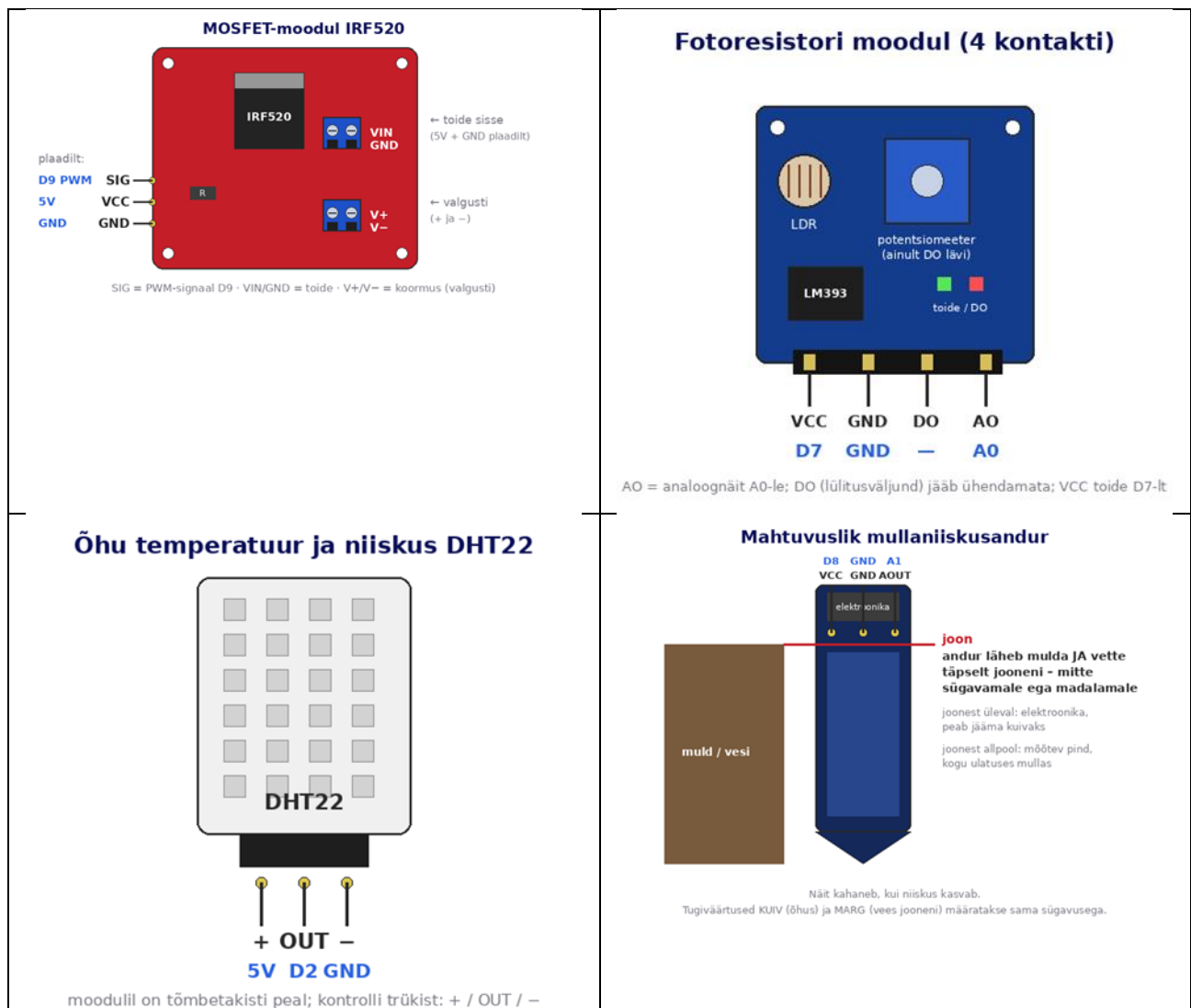
Valgusti nimipinge on 3 V, IRF520 lülitab aga plaadi 5 V toidet. Heleduse juhtimine impulsslaiusmodulatsiooniga (ingl pulse-width modulation, PWM) ei alanda pinget – iga impulsi ajal saab valgusti täis 5 V –, seepärast on valgustiga jadamisi eeltakisti, mis piirab voolu. Takisti arvutatakse valgusti nimivoolu I järgi:

$$R = (5\text{ V} - 3\text{ V}) / I, \text{ takisti võimsus } P = (5\text{ V} - 3\text{ V}) \cdot I.$$

Näiteks 20 mA voolu korral on $R = 100\ \Omega$ ja $P = 0,04$ W, 60 mA korral $R \approx 33\ \Omega$ ja $P = 0,12$ W; mõlemal juhul sobib 0,25 W takisti. Kui nimivool pole teada, mõõdetakse see multimeetriga, kui valgusti töötab oma toiteallikaga (nt kaks AA-elementi). Takisti joodetakse valgusti plussjuhtmesse ja kaetakse kuumakahaneva toruga. USB-pistikuga valgusti on 5 V seade: sellisel juhul eeltakistit ei lisata.



Joonis 2. Arduino UNO R4 WiFi – kasutatavad kontaktid

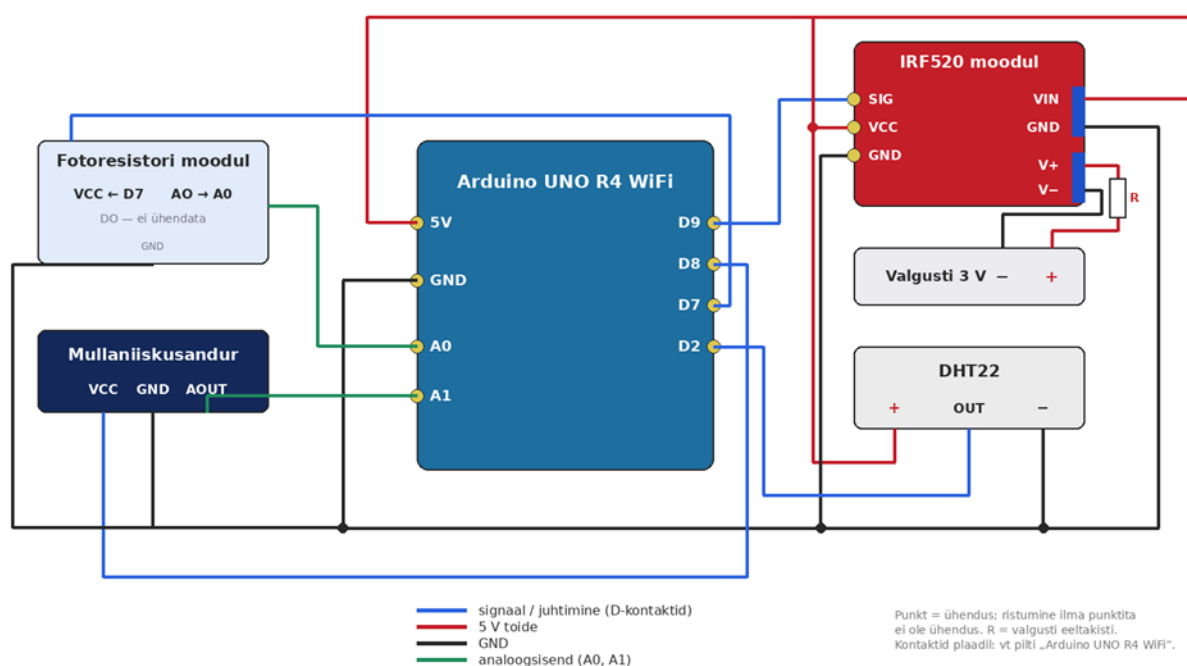


Joonis 3. Moodulid ja nende kontaktid: MOSFET-moodul IRF520, fotoresistori moodul, DHT22 ja mullaniiskusedur

3 Kokkupanek

Moodul ja kontakt	Plaadi kontakt	Selgitus
Valgusmoodul VCC / GND / AO	D7 / GND / A0	toide D7-lt: moodul on sees ainult mõõtmise ajal
Mullaandur VCC / GND / AOUT	D8 / GND / A1	toide D8-lt: pidevalt pinge all olev andur korrodeerub mullas kiiremini
DHT22 + / - / OUT	5V / GND / D2	lugemine kõige tihedamalt iga 2 s järel
IRF520 SIG / VCC / GND	D9 / 5V / GND	D9 annab PWM-signaali (heledus)
IRF520 VIN / GND (kraviklemm)	5V / GND	valgusti toide läbi mooduli
IRF520 V+ → eeltakisti → valgusti +	—	eeltakisti on alati V+ ja valgusti vahel
IRF520 V- → valgusti -	—	kui valgusti ei sütti, kontrolli polaarsust

Terraariumi seadme ühendusskeem



Joonis 4. Ühendusskeem: punkt tähistab ühendust, ristumine ilma punktita ei ole ühendus; R on valgusti eeltakisti

Laul kasutatakse maketeerimisplaati ja Dupont-juhtmeid. Terraariumis, kus õhuniiskus on pidevalt kõrge, on kolm sagedasemat rikkekohta: jootekohad andurite kontaktribadel, oksüdeeruvad Dupont-otsad ja niiskus, mis liigub piki juhtmesooni elektroonikani. Püsipaigalduseks joodetakse juhtmed või kaetakse ühenduskohad vähemalt kuumakahaneva toruga ning plaat ja moodulid paigutatakse niiskuskindlasse karpi anuma kõrvale.

Anumasse lähevad ainult mullaandur (mulda täpselt trükitud jooneni) ja DHT22 (ripub anuma kaela all, 5 cm korgist allpool). Valgusmoodul on anumast väljas korgi peal, fotoresistor (ingl light-dependent resistor, LDR) suunaga lakke: see mõõdab ruumi valgust, mitte terraariumi oma, ja valgusti valgus sellele ei lange. Juhtmed viiakse läbi korgi 8 mm ava, mis tihendatakse pärast

paigaldamist niiskuskindlalt; tihendamise viisi valib tegija. Täpne paigutus anumas on terraariumi kokkupaneku tööjuhendis (peatükid 4 ja 6).

4 Pilvepool: tabel ja veebirakendus näidis: Lisa B

Pilvepool tehakse üks kord kõigi seadmete jaoks ja seda katsetatakse brauseris enne, kui ükski plaat sinna midagi saadab.

Samm	Toiming	Kontroll
1	Loo Google Drive'is uus pilvetabel ja nimeta see „Terraarium“. File → Settings → Time zone: vali Tallinn → Save settings. Kopeeri aadressiribalt tabeli ID: osa /spreadsheets/d/ ja /edit vahel.	ID on ~44 märki; ajavöönd on Tallinn
2	Extensions → Apps Script. Kustuta avanenud faili Code.gs kogu sisu ja kleebi asemele Lisa B. Real const TABEL_ID = "SIIA_TABELI_ID" asenda jutumärkide vahel olev tekst tabeli ID-ga. Salvesta (Ctrl+S).	projekt salvestub ilma punase veata
3	Vali funktsioonide loendist seadistaTabel, vajuta Run. Esimesel korral küsib Google luba: Review permissions → oma konto → Allow; kui kuvatakse hoiatus „Google hasn't verified this app“, siis Advanced → Go to ... (unsafe) → Allow.	tabelisse tekivad lehed Andmed, Seadmed, Konfig; Konfigi veerus led_reziim on rippmenüü; Apps Scripti vasakus menüüs Triggers (kell) on rida arhiveeriAndmed
4	Vali testiSalvestamist, Run.	lehel Andmed on rida T00; lehel Seadmed rida T00 staatusega OK; lehel Konfig rida T00 vaikeväärtustega
5	Deploy → New deployment → tüüp Web app; Execute as: Me; Who has access: Anyone. Deploy. Kopeeri Web app URL (veebiaadress, ingl Uniform Resource Locator).	URL algab https://script.google.com/macros/s/ ja lõpeb /exec; kui valikut Anyone ei ole, peab selle lubama Google Workspace'i administraator
6	Brauseris: <URL>?action=saada&id=T99&temp=21.5&niiskus=55&muld=40&valgus=300&led=kustu	vastus {"ok":true,...}; uus rida lehel Andmed
7	Brauseris: <URL>?action=konfig&id=T99	vastus {"ok":true,"konfig":{"...}}; konfig-osas on 14 välja
8	Kustuta katseread T00 ja T99 lehtedelt Andmed, Seadmed ja Konfig (rea number → Delete row).	lehtedel on ainult päised
9	Iga hilisema koodimuudatuse järel: Deploy → Manage deployments → pliiats → Version: New version → Deploy.	URL jääb samaks; ilma uue versioonita töötab vana kood edasi

Veebirakenduse aadressi käsitletakse nagu parooli: kes aadressi teab, saab tabelisse ridu lisada ja seadmete seadeid lugeda. Seepärast ei avaldata aadressi õppematerjalides ega veebilehtedel. Pärast sammu 8 on pilvepool valmis; lehe Konfig sisu ja tähendus on peatükis 6, kuuarhiiv peatükis 7.

5 Plaadi programm näidis: Lisa A

Samm	Toiming	Kontroll
1	Paigalda arenduskeskkond Arduino IDE 2 (ingl Integrated Development Environment; arduino.cc/en/software). Ühenda plaat USB-C andmekaabliga.	Tools → Port näitab porti
2	Tools → Board → Boards Manager, otsi „UNO R4“, paigalda „Arduino UNO R4 Boards“. Tools → Board → Arduino UNO R4 WiFi.	plaadi nimi akna alumises servas
3	Tools → Manage Libraries: paigalda „DHT sensor library“ (Adafruit; koos „Adafruit Unified Sensor“) ja „ArduinoJson“ (Benoit Blanchon). ArduinoJsoni puhul vali versioonide rippmenüüst viimane 6.x versioon, sest vaikimisi pakutakse versiooni 7. WiFiS3 ja EEPROM tulevad plaadipaketiga.	kolm teeki loendis „Installed“, ArduinoJson 6.x
4	File → New Sketch, kustuta avanenud faili kogu sisu ja kleebi asemele Lisa A. Täida jaotises KÕVAKOODITUD neli rida: SSID (võrgu nimi), PAROOL, SEADE_ID (nt T01) ja WEBAPP_PATH (peatüki 4 sammus 5 kopeeritud URL-i osa alates /macros/s/ kuni /exec, ilma osata https://script.google.com).	muutuvad ainult need neli rida
5	Verify (linnuke); kui vigu pole, Upload (nool).	„Done compiling“, seejärel „Done uploading“; vea korral vt peatükk 9
6	Tools → Serial Monitor, kiirus 9600. Vajuta plaadil RESET.	jadapordis „>> WiFi OK“, „>> Konfig OK“ ja „>> Valmis. Esimene saatmine ... s pärast“ (T01 puhul 30 s, iga järgmise numbri puhul 10 s rohkem); mõõterida iga 10 s järel; pärast saatmist rida „>> Vastus: ...“, milles on "ok":true
7	Pilvetabel: lehel Andmed rida seadme ID-ga; lehel Seadmed rida uuenenud; lehel Konfig rida tekkinud.	kolm lehte

Plaadi olekutuli L näitab olekut ka ilma arvutita. Pärast käivitust tähendab 3 lühikest vilget, et WiFi-ühendus on olemas, ja 10 vilget, et ühendus ebaõnnestus. Iga saatmise järel tähendab 2 vilget, et

veebirakendus kinnitas rea salvestamist, ja 5 vilget, et saatmine ebaõnnestus (ka siis, kui WiFi-ühendus kaob töö ajal) või DHT22 ei vasta.

Jadapordi väljundi read algavad märgistusega: [MÕÕT] – näidud, >> – tegevus, ! – hoiatus (programm parandab väärtuse või kasutab ohutut olekut), X – viga. Vea korral kopeeritakse keelemudelile X- ja !-read koos neile eelnenud >>-ridadega. Programm ei kirjuta jadaporti võrgu nime, IP-aadressi (internetiprotokolli aadress, ingl Internet Protocol) ega veebirakenduse aadressi.

6 Seadistamine ja kalibreerimine leht Konfig

Lehel Konfig on iga seadme kohta üks rida ja see on lahenduse haldusliides. Veerus A on seadme ID, seaded on veergudes B–O allolevas järjekorras. Muudatus jõuab seadmeni järgmise konfigipäringuga, hiljemalt konfig_min minuti pärast.

Veerg	Tähendus	Vaikeväärtus
B, C: syttib, kustub	valguse näit, mille juures valgusti automaatrežiimis süttib ja kustub; näit kasvab pimedas; syttib on suurem kui kustub, vahe vähemalt 30	190, 150
D: heledus	valgusti heledus 0–255	150
E, F: kuiv, marg	mullaanduri toornäit kuivalt õhus ja vees jooneni	620, 300
G: saada_min	saatmise vahe minutites	5
H–M: temp_min ... muld_max	temperatuuri, õhuniiskuse ja mulla niiskuse piirid, millest väljas on lehel Seadmed staatus HOIATUS koos põhjusega	10–35 °C, 30–100 %, 15–90 %
N: led_reziim	auto / sees / kustus (rippmenüüst)	auto
O: konfig_min	seadete küsimise vahe minutites	5

Kui syttib ja kustub vahe on väiksem kui 30, laiendab programm vahet ise ja kirjutab jadaporti hoiatuse, kuid lehel Konfig jääb sisestatud väärtus alles. Seepärast parandatakse väärtus ka tabelis, et tabel ja seade oleksid kooskõlas. Heleduse väärtused alla 0 loeb programm väärtuseks 0 ja üle 255 väärtuseks 255.

Kalibreerimise järjekord

Mullaanduri kalibreerimine tehakse enne, kui andur mulda paigaldatakse. Jadapordist loetakse mullaanduri toornäit kuivalt õhus (kuiv) ja klaasis vees täpselt trükitud jooneni (marg) ning kantakse lehele Konfig. Seejärel loetakse valguse näit ruumi tavavalguses ja hämaras. Kuna näit kasvab pimedas, valitakse kustub veidi suurem kui tavavalguse näit ja syttib veidi väiksem kui hämara näit, nii et syttib ja kustub vahe on vähemalt 30.

Valgusti automaatrežiim

Vaikimisi on valgusti automaatrežiimis ja lülitub ruumi valguse järgi: kui ruum on pime, põleb valgusti, sealhulgas öösel, nädalavahetusel ja koolivaheajal. Kui taimedele on vaja kindla pikkusega valgusperioodi ja pimedat aega, kasutatakse režiime sees ja kustus või lisatakse programmi ajaaken peatüki 9 neljanda näidisküsimuse järgi.

Õhuniiskuse mõõtmine

Suletud terraariumis on suhteline õhuniiskus sageli 85–100 %. See on DHT22 mõõtepiirkonna ülemine osa, kus mõõteviga on suurem ja kondensaat võib näitu ajutiselt moonutada. Õhuniiskuse näitu kasutatakse seetõttu eelkõige muutuste jälgimiseks. Seepärast on `niiskus_max` vaikeväärtus 100 %, et suletud terraariumi tavaline niiskus ei annaks püsivat hoiatust; `niiskus_min` vaikeväärtusega 30 % tuleb hoiatus, kui terraarium on kuivanud või jäänud lahti.

7 Mitme seadme tootmine

Kõik seadmed kasutavad sama veebirakendust ja sama programmi; erinev on ainult SEADE_ID. ID-skeem on T01 ... T30 (täht T ja kahekohaline number), sama ID kleebitakse sildina plaadile ja terraariumile. Numbri järgi nihutab programm ka saatmise aega: T01 saadab esimest korda 30 s pärast jadapordi teadet „>> Valmis“, T02 40 s pärast ja nii edasi kuni T30-ni (320 s), nii et seadmed ei saada pärast ühist volukatkestust samal hetkel. Lehel Seadmed kirjutatakse veergu Asukoht ruum või klass, mitte isiku nimi. Konfigi rida tekib igale seadmele automaatselt esimese saatmisega; kalibreerimisväärtused kuiv, marg, syttib ja kustub on seadme põhised ja sisestatakse peatüki 6 järgi.

Seadme kontroll enne kasutuselevõttu	✓
ID programmis, plaadil ja tabelis on sama	☐
Jadaport: „>> WiFi OK“, „>> Konfig OK“, mõõterida iga 10 s järel ilma küsimärgideta, pärast saatmist rida „>> Vastus: ...“, milles on "ok":true	☐
Pilvetabel: rida lehel Andmed ja lehel Seadmed staatus OK	☐
Valgusti: led_reziim = sees → põleb; kustus → ei põle; auto → lülitub 20–30 s jooksul, kui LDR kinni katta	☐
Konfigi muudatus (nt heledus) jõuab seadmeni konfig_min minuti jooksul	☐
Toide lahti ja tagasi: režiimis sees või kustus on valgusti kohe õiges olekus, režiimis auto umbes 30 s jooksul, kui WiFi-võrk on olemas, ja kuni 2 min jooksul ilma võrguta (püsimalu)	☐
Mullaandur mullas täpselt jooneni, DHT22 5 cm korgist allpool, valgusmoodul korgi peal LDR suunaga lakke, juhtmed kinnitatud	☐
Korgi juhtmeava tihendatud	☐
24 h tabelis: saada_min = 5 juures umbes 288 rida, pikim vahe alla 10 min	☐

Pärast esimest töötavat seadet kulub ühe seadme peale umbes 20–30 minutit: kokkupanek (peatükk 3), üleslaadimine (peatükk 5, sammud 4–7), kalibreerimine (peatükk 6) ja kontrollnimekiri; 24 tunni kontroll kulgeb samal ajal järgmiste seadmete tootmisega.

Kolmkümmend seadet tekitavad saada_min = 5 juures 8640 rida ööpäevas ehk 30-päevases kuus 259 200 rida. Pilvetabeli ühes failis võib olla kuni 10 miljonit lahtrit ja arvesse lähevad ka lehe tühjad veerud, seepärast on leht Andmed ainult seitsme veeru laiune ja kuu andmed võtavad umbes 1,8 miljonit lahtrit. Iga kuu 1. päeval kella 1 paiku kopeerib veebirakendus lehe Andmed uude faili Terraarium_arhiiv_AAAA_KK (sama konto Google Drive'is, kaustas Minu Drive) ja kustutab kopeeritud read; seadmed saadavad samal ajal edasi ja ükski rida ei kao ega satu arhiivi kaks korda.

Arhiveerimise saab käivitada ka käsitsi: Apps Scripti redaktoris vali arhiveeriAndmed ja vajuta Run. Kui automaatne arhiveerimine jääb mingil põhjusel ära, mahuvad tabelisse umbes viie kuu andmed; päästiku olemasolu näeb Apps Scripti vasakust menüüst Triggers.

8 Hooldus ja tüüpilised vead

Igapäevane hooldus on lehe Seadmed vaatamine. Staatus näitab, milline näit on piiridest väljas; seadme töötamist näitab veerg „Viimane uuendus“. Seade, mis ei saada, oma staatust ei muuda. Kui „Viimane uuendus“ on vanem kui saada_min + 10 minutit, ei jõua seadme andmed tabelisse: põhjus võib olla võrguühenduses, toites või anduris DHT22 (kui DHT22 ei vasta, jätab plaat saatmise vahele). Seadeid muudetakse lehel Konfig, plaadi juurde minemata.

Sümptom	Tõenäoline põhjus	Kontroll
Port puudub arenduskeskkonnas	laadimiskaabel ilma andmesoonteta; USB-pesa	teine kaabel
Jadapordis arusaamatud märgid	vale kiirus	Serial Monitor 9600
Temperatuur ja niiskus „?“	DHT22 juhe või toide; andur rikkis	D2 ühendus, + ja – kontaktid
Valgus alati 0 või 1023	juhe DO-s, mitte AO-s; D7 ei ole lugemise ajal sees	AO kontakt
Muld alati sama number	D8 toide ei ole sees; AOUT vales kontaktis	andur pihku – näit peab muutuma
„X IP-aadress puudub“	võrk ei jaganud aadressi (dünaamilise hostikonfiguratsiooni protokoll, ingl Dynamic Host Configuration Protocol, DHCP)	programm proovib ise uuesti; kas võrgus on vabu aadresse
„X WiFi ebaõnnestus“	vale võrgu nimi või parool; võrk on ainult 5 GHz; seade on levialast väljas	SSID ja PAROOL; 2,4 GHz võrk

Sümptom	Tõenäoline põhjus	Kontroll
„X HTTPS ühendus ebaõnnestus“	kui eelneb „X WiFi ebaõnnestus“, puudub WiFi-ühendus; muidu nõuab võrk sisselogimislehte või tulemüür piirab ühendust	teine 2,4 GHz võrk (nt telefoni jagatud võrk)
Esimene WiFi-katse ebaõnnestub, teine õnnestub	raadiomoodul ei käivitunud puhtalt	tavaline; programm proovib 3 korda
„X Redirect puudub“	vale WEBAPP_PATH	brauseris sama aadress
„X Sisselogimisleht: juurdepääs ei ole Anyone“	veebirakenduse juurdepääs on piiratud	Deploy → Manage deployments: Who has access: Anyone
„X Tabel ei salvestanud rida“	veebirakendus vastas ok:false: lukk hõivatud või viga tabelis	jadapordis „>> Vastus“ ja selle error-väli
Brauserist tuleb rida, plaadilt ei tule	vana deployment-aadress; parameetri nimi erineb	võrdle päringut täht-tähelt
Konfig ei muutu	plaat kasutab püsimalu, pilvepäring ebaõnnestub	kas jadapordis on „>> Konfig OK“ iga konfig_min järel?
Valgusti lülitub sageli sisse ja välja	syttib/kustub on ruumi tavavalguse näidule liiga lähedal; LDR-ile langeb valgusti või akna valgus	LDR suund; syttib ja kustub uuesti (peatükk 6)
Valgusti põleb öösel	automaatrežiim lülitab ruumi valguse järgi	peatükk 6, valgusti automaatrežiim
Valgusti ei sütti	VIN/GND toide puudub; V+ ja V– juhtmed omavahel vahetatud; eeltakisti katki; seaded laadimata (ohutu olek = kustus)	kas jadapordis on „! EEPROM tühi, LED kustus kuni pilvekonfig saabub“ ja puudub „>> Konfig OK“; polaarsus; takisti
Staatust „HOIATUS: õhuniiskus“ püsivalt	suletud terraariumis on 85–100 % tavaline; niiskus_max on väiksem kui 100	niiskus_max = 100 (peatükk 6)
Seade töötab laual, terraariumis rikneb päevadega	niiskus ühendustes	jootmine, kuumakahanev toru, karp, korgi tihendus

9 Keelemudel muudatuste ja vigade otsimisel

Valmis lahenduse juures on suur keelemudel (SKM) muudatuste ja vigade otsimise tööriist, mitte programmi allikas – programm on Lisa A ja Lisa B. Sobivad kõik suuremad üldotstarbelised keelemudelid: kooli kontoga Microsoft Copilot, tasuta tasemel ChatGPT (OpenAI), Claude

(Anthropic) ja Gemini (Google). Vastuse kvaliteet sõltub peamiselt sellest, kui täpselt on kirjeldatud plaat, teegid, ühendused ja soovitud tulemus; keelemudelile antakse alati kogu programm või kogu veateade, mitte ümberjutustus.

Kui keelemudelile antakse oma, juba täidetud programm, asendatakse enne kleepimist salajased väärtused samade kohatäidetega nagu lisades: võrgu nimi → VÕRGU_NIMI, WiFi parool → VÕRGU_PAROOL, veebirakenduse aadressi osa /macros/s/ ja /exec vahel → DEPLOYMENT_ID ning tabeli ID → SIIA_TABELI_ID. Lisades A ja B on need kohatäited juba olemas.

Neli tüüpolukorda koos küsimusega, mis on andnud kasutatava vastuse:

Näidisküsimus keelemudelile. See on Arduino UNO R4 WiFi programm (kleebi Lisa A). Verify annab vea (kleebi punane tekst tervikuna). Ütle, mis on põhjus, ja näita parandatud rida koos kahe eelneva ja kahe järgneva reaga – ära kirjuta kogu programmi uuesti.

Näidisküsimus keelemudelile. See on töötav programm (kleebi Lisa A). Lisa BME280 andur I²C-siini (ingl Inter-Integrated Circuit) kaudu (Qwiic-pesa, teek Adafruit BME280) nii, et temperatuur ja niiskus tuleksid sealt ja DHT22 jääks varuks: kui BME280 ei vasta, kasutatakse DHT22. Muuda ainult vajalikke kohti ja märgi iga muudetud rida kommentaariga „// UUS“. Anna lõpus loetelu, mida Library Managerist juurde paigaldada.

Näidisküsimus keelemudelile. See on Apps Scripti kood (kleebi Lisa B). Lisa lehele Andmed veerg „Aku V“ ja võta parameeter aku vastu samamoodi nagu temp. Vanad read, kus aku puudub, peavad jääma toimima. Näita ainult muudetud funktsioone.

Näidisküsimus keelemudelile. See on töötav programm (kleebi Lisa A) ja selle Apps Scripti pool (kleebi Lisa B). Lisa valgustile ajaaken: automaatrežiimis tohib valgusti põleda ainult täistundide vahemikus, mis on lehel Konfig uutes veergudes valgus_algus ja valgus_lopp (Eesti aeg). Kellaaeg võetakse võrgust ja hoitakse plaadi reaajakellas. Muuda ainult vajalikke kohti, märgi muudetud read kommentaariga „// UUS“ ja näita eraldi Lisa B muudatused.

Keelemudeli vastust ei võeta kasutusele enne, kui see on ühel seadmel jadapordi ja pilvetabeli järgi üle kontrollitud; alles siis läheb sama programm teistele plaatidele. Kui keelemudel pakub lahendust, mis muudab tabeli veergude järjekorda või päringu parameetrite nimesid, tuleb muuta ka teist poolt – plaat ja veebirakendus on kokku lepitud liides, mida muudetakse alati koos.

Lisa A – plaadi näidisprogramm Terraarium_naidis.ino

Kasutamine: peatükk 5, samm 4. Muuta tuleb ainult nelja rida jaotises KÕVAKOODITUD: SSID, PAROOL, SEADE_ID ja WEBAPP_PATH; ülejäänud seaded tulevad lehelt Konfig (peatükk 6). Vajalikud teegid: WiFiS3 ja EEPROM (plaadipaketiga), DHT sensor library koos teegiga Adafruit Unified Sensor ning ArduinoJson 6 (peatükk 5, samm 3).

```
// =====  
// Terraarium -> pilvetabel . näidisprogramm v2.3  
// v2.1: valguse keskmistamine, lülituse viivitus, min-hüsterees,  
//      konfig EEPROM-is (pärast volukatkestust LED õiges režiimis)  
// v2.2: heledus piiratud 0-255, led-väärtus ASCII-s (poleb/kustu),  
//      led_reziim väiketähtedeks, kasutamata funktsioonid eemaldatud  
// v2.3: saatmise nihe seadme numbri järgi, salvestuse kontroll  
//      (ok:true), sisselogimislehe tuvastus, niiskus_max 100  
// =====  
// Teegid: WiFiS3 ja EEPROM (plaadipaketiga), DHT sensor library,  
//      Adafruit Unified Sensor, ArduinoJson 6  
// =====  
  
#include <WiFiS3.h>  
#include <DHT.h>  
#include <ArduinoJson.h>  
#include <EEPROM.h>  
  
// ===== KÕVAKOODITUD =====  
// 2,4 GHz võrgu nimi (SSID) ja parool  
const char* SSID      = "VÕRGU_NIMI";  
const char* PAROOL    = "VÕRGU_PAROOL";  
// igal seadmel oma ID (T01 ... T30)  
const char* SEADE_ID = "T01";  
  
const char* HOST = "script.google.com";  
// veebirakenduse aadressi osa alates /macros/s/ kuni /exec  
// (ilma https://script.google.com osata)  
const char* WEBAPP_PATH = "/macros/s/DEPLOYMENT_ID/exec";  
  
// ===== KONTAKTID =====  
const int valgusPin   = A0;  
const int muldPin     = A1;  
const int dhtPin      = 2;  
const int ledPin      = 9;  
const int statusLed   = LED_BUILTIN;  
const int valgusToide = 7;  
const int muldToide   = 8;  
  
DHT dht(dhtPin, DHT22);  
  
// andurite toite stabiliseerumise aeg enne lugemist (ms)  
const int MULD_STABIIL = 500;  
const unsigned long LED_KONTROLL_VAHE = 10000;  
  
// --- valgusti lülituse kaitse ---  
// mitu analogRead'i keskmistatakse  
const int VALGUS_PROOVE = 16;  
// mitu järjestikust kontrolli enne lülitust (3 × 10 s)  
const int LED_KINNITUSI = 3;  
// syttib - kustub ei tohi olla väiksem  
const int MIN_HYSTEREES = 30;  
  
// ===== KONFIG (pilvest) =====  
// vaikeväärtused = Lisa B VAIKIMISI  
int SYTTIB          = 190;
```

```

int KUSTUB          = 150;
int HELEDUS         = 150;
int KUIV            = 620;
int MARG            = 300;
int SAADA_MIN       = 5;
int KONFIG_MIN      = 5;
float TEMP_MIN      = 10;
float TEMP_MAX      = 35;
float NIISKUS_MIN   = 30;
float NIISKUS_MAX   = 100;
float MULD_MIN      = 15;
float MULD_MAX      = 90;
String LED_REZIIM   = "auto";

unsigned long saada_vahe_ms = 5UL * 60UL * 1000UL;
unsigned long konfig_vahe_ms = 5UL * 60UL * 1000UL;

// ===== MUUTUJAD =====
int valgus          = 0;
int muldToor        = 0;
int muldProtsent    = 0;
bool poleb          = false;
float temp           = NAN;
float niiskus        = NAN;

// järjestikuste "vaheta olekut" kontrollide loendur
int ledKinnitus     = 0;
// false -> valgusti hoitakse kustus (ohutu olek)
bool konfiglaetud   = false;
// konfigi uus katse iga 30 s, kuni konfig pole laetud
const unsigned long KONFIG_RETRY_MS = 30000;

// ---- EEPROM-i struktuur ----
// "TER1"
const uint32_t EEPROM_MAGIC = 0x54455231;
struct KonfigSalv {
    uint32_t magic;
    int syttib, kustub, heledus, kuiv, marg, saada_min, konfig_min;
    float temp_min, temp_max, niiskus_min, niiskus_max;
    float muld_min, muld_max;
    char led_reziim[12];
};

unsigned long viimaneLedKontroll = 0;
unsigned long viimaneSaada       = 0;
unsigned long viimaneKonfig      = 0;

// esimese saatmise nihe seadme numbri järgi (arvutatakse setup-is)
unsigned long nihe_ms = 30000;
bool esimeneSaatmine = true;
bool esimeneKonfig   = true;

WiFiSSLClient klient;

// =====
void setup() {
    pinMode(ledPin, OUTPUT);
    pinMode(statusLed, OUTPUT);
    digitalWrite(statusLed, LOW);

    pinMode(valgusToide, OUTPUT);
    pinMode(muldToide, OUTPUT);
    digitalWrite(valgusToide, LOW);
    digitalWrite(muldToide, LOW);

    Serial.begin(9600);
    unsigned long t0 = millis();
    while (!Serial && millis() - t0 < 3000) {}

```

```

Serial.println();
Serial.println("=== Terraarium · seireseade v2.3 ===");
Serial.print("Seade: "); Serial.println(SEADE_ID);

// T01 saadab 30 s, T02 40 s ... T30 320 s pärast käivitust,
// et seadmed ei saadaks pärast voolukatkestust samal hetkel
int nr = atoi(SEADE_ID + 1);
if (nr < 1) nr = 1;
nihe_ms = 30000UL + (unsigned long)((nr - 1) % 30) * 10000UL;

dht.begin();
delay(200);

// viimane teadaolev konfiguratsioon EEPROM-ist: valgusti õiges režiimis
// ka ilma võrguta
if (laeKonfigEEPROM()) {
    Serial.print(">> EEPROM konfiguratsioon: LED režiim = ");
    Serial.println(LED_REZIIM);
} else {
    Serial.println("! EEPROM tühi, LED kustus kuni pilvekonfiguratsioon saabub");
}
juhiLed();

if (yhendaWifi()) {
    vilgutaStatus(3, 100);
    laadiKonfig();
    loeKoikAndurid();
    naitaMooted();
} else {
    vilgutaStatus(10, 80);
}
Serial.print(">> Valmis. Esimene saatmine ");
Serial.print(nihe_ms / 1000); Serial.println(" s pärast");
viimaneSaada = millis();
viimaneKonfig = millis();
}

// =====
void loop() {

    // iga 10 s: loe kõik andurid ja näita jadapordis
    if (millis() - viimaneLedKontroll >= LED_KONTROLL_VAHE) {
        viimaneLedKontroll = millis();
        loeKoikAndurid();
        juhiLed();
        naitaMooted();
    }

    // käsitsirežiim (sees/kustus) rakendub kohe
    if (LED_REZIIM != "auto") {
        juhiLed();
    }

    // kuni pilvekonfiguratsioon pole kordagi laetud: WiFi + konfiguratsioon iga 30 s
    if (!konfigLaetud && millis() - viimaneKonfig >= KONFIG_RETRY_MS) {
        viimaneKonfig = millis();
        if (WiFi.status() != WL_CONNECTED) yhendaWifi();
        if (WiFi.status() == WL_CONNECTED) laadiKonfig();
    }

    // esimene korraline konfiguratsioon 5 s pärast esimest saatmist, siis
    // iga konfiguratsioon järel (seadmete päringud jäävad nihkesse)
    unsigned long konfigOota =
        esimeneKonfig ? nihe_ms + 5000UL : konfig_vahe_ms;
    if (konfigLaetud && millis() - viimaneKonfig >= konfigOota) {
        viimaneKonfig = millis();
        esimeneKonfig = false;
    }
}

```

```

    if (WiFi.status() == WL_CONNECTED) laadiKonfig();
}

unsigned long saadaOota =
    esimeneSaatmine ? nihe_ms : saada_vahe_ms;
if (millis() - viimaneSaada >= saadaOota) {
    viimaneSaada = millis();
    esimeneSaatmine = false;
    teeMooteSjaSaada();
}
}

// =====
// Loeb kõik andurid korraga (valgus + muld + DHT22)
// =====
void loeKoikAndurid() {
    digitalWrite(valgusToide, HIGH);
    digitalWrite(muldToide, HIGH);
    delay(MULD_STABIIL);

    valgus = loeValgusKeskmine();
    muldToor = analogRead(muldPin);
    temp = dht.readTemperature();
    niiskus = dht.readHumidity();

    digitalWrite(valgusToide, LOW);
    digitalWrite(muldToide, LOW);

    muldProtsent = map(muldToor, KUIV, MARG, 0, 100);
    muldProtsent = constrain(muldProtsent, 0, 100);
}

// keskmistatud valgusnäit; toide peab olema juba sees
int loeValgusKeskmine() {
    // esimene näit jäetakse kõrvale (ADC stabiliseerumine)
    analogRead(valgusPin);
    long summa = 0;
    for (int i = 0; i < VALGUS_PROOVE; i++) {
        summa += analogRead(valgusPin);
        delay(2);
    }
    return (int)(summa / VALGUS_PROOVE);
}

void juhiLed() {
    // konfiguratsioon -> valgusti kustus
    if (!konfigLaetud) {
        poleb = false;
        analogWrite(ledPin, 0);
        return;
    }
    if (LED_REZIIM == "sees") {
        poleb = true;
        analogWrite(ledPin, HELEDUS);
    } else if (LED_REZIIM == "kustus") {
        poleb = false;
        analogWrite(ledPin, 0);
    } else {
        // auto: lülitub alles siis, kui LED_KINNITUSI järjestikust
        // kontrolli näitavad sama suunda; üksik kõikumine ei lülita
        bool tahabPolema = !poleb && valgus > SYTTIB;
        bool tahabKustuda = poleb && valgus < KUSTUB;
        if (tahabPolema || tahabKustuda) {
            ledKinnitus++;
            if (ledKinnitus >= LED_KINNITUSI) {
                poleb = !poleb;
                ledKinnitus = 0;
            }
        }
    }
}

```

```

    } else {
        ledKinnitus = 0;
    }
    analogWrite(ledPin, poleb ? HELEDUS : 0);
}
}

// =====
void teeMooteSjaSaada() {
    Serial.println();
    Serial.println(">> Saatmise aeg");
    loeKoikAndurid();
    juhiLed();
    naitaMoote();

    if (isnan(temp) || isnan(niiskus)) {
        Serial.println("X DHT22 ei vasta, saatmine jääb vahele");
        vilgutaStatus(5, 100);
        return;
    }

    digitalWrite(statusLed, HIGH);
    if (WiFi.status() != WL_CONNECTED) yhendaWifi();
    bool ok = saadaAndmed(temp, niiskus, muldProtsent, valgus, poleb);
    digitalWrite(statusLed, LOW);

    if (ok) vilgutaStatus(2, 150);
    else vilgutaStatus(5, 100);
}

void naitaMoote() {
    Serial.print("[MÕÖT] valgus: "); Serial.print(valgus);
    Serial.print(" LED:");
    Serial.print(poleb ? "poleb" : "kustu");
    Serial.print(" ("); Serial.print(LED_REZIIM); Serial.print(")");
    Serial.print("\ttemp: ");
    Serial.print(isnan(temp) ? "?" : String(temp, 1));
    Serial.print(" °C\ttõhk: ");
    Serial.print(isnan(niiskus) ? "?" : String(niiskus, 0));
    Serial.print(" %\tmuld: "); Serial.print(muldProtsent);
    Serial.print(" % (toor "); Serial.print(muldToor);
    Serial.println(")");
}

// =====
bool laadiKonfig() {
    Serial.println(">> Laen konfigi pilvest");

    String tee = String(WEBAPP_PATH) + "?action=konfig&id="
        + String(SEADE_ID);
    String keha = "";
    if (!teeGetParing(HOST, tee, keha)) {
        Serial.println("X Konfig ei tulnud");
        return false;
    }

    int jStart = keha.indexOf('{');
    int jEnd = keha.lastIndexOf('}');
    if (jStart < 0 || jEnd <= jStart) {
        Serial.println("X JSON-i ei leitud vastusest");
        return false;
    }
    String jsonOsa = keha.substring(jStart, jEnd + 1);
    Serial.print(">> JSON: "); Serial.println(jsonOsa);

    StaticJsonDocument<1024> doc;
    DeserializationError err = deserializeJson(doc, jsonOsa);
    if (err) {

```

```

    Serial.print("X JSON viga: "); Serial.println(err.c_str());
    return false;
}
if (!doc["ok"]) { Serial.println("X ok:false"); return false; }

JsonObject k = doc["konfig"];
if (k.containsKey("syttib")) SYTTIB = k["syttib"];
if (k.containsKey("kustub")) KUSTUB = k["kustub"];
if (k.containsKey("heledus")) HELEDUS = k["heledus"];
if (k.containsKey("kuiv")) KUIV = k["kuiv"];
if (k.containsKey("marg")) MARG = k["marg"];
if (k.containsKey("saada_min")) {
    SAADA_MIN = max(1, (int)k["saada_min"]);
    saada_vahe_ms = (unsigned long)SAADA_MIN * 60UL * 1000UL;
}
if (k.containsKey("konfig_min")) {
    KONFIG_MIN = max(1, (int)k["konfig_min"]);
    konfig_vahe_ms = (unsigned long)KONFIG_MIN * 60UL * 1000UL;
}
if (k.containsKey("temp_min")) TEMP_MIN = k["temp_min"];
if (k.containsKey("temp_max")) TEMP_MAX = k["temp_max"];
if (k.containsKey("niiskus_min")) NIISKUS_MIN = k["niiskus_min"];
if (k.containsKey("niiskus_max")) NIISKUS_MAX = k["niiskus_max"];
if (k.containsKey("muld_min")) MULD_MIN = k["muld_min"];
if (k.containsKey("muld_max")) MULD_MAX = k["muld_max"];
if (k.containsKey("led_reziim")) {
    LED_REZIIM = String((const char*)k["led_reziim"]);
}
kontrolliKonfig();

konfigLaetud = true;
salvestaKonfigEEPROM();

Serial.print(">> Konfig OK. LED režiim: "); Serial.print(LED_REZIIM);
Serial.print(", saadan iga "); Serial.print(SAADA_MIN);
Serial.print(" min, konfig iga "); Serial.print(KONFIG_MIN);
Serial.println(" min");
return true;
}

// väärtuste kontroll: heledus 0-255, režiim väiketähtedega,
// hüsterees vähemalt MIN_HYSTEREES
void kontrolliKonfig() {
    HELEDUS = constrain(HELEDUS, 0, 255);
    LED_REZIIM.trim();
    LED_REZIIM.toLowerCase();
    if (LED_REZIIM != "auto" && LED_REZIIM != "sees"
        && LED_REZIIM != "kustus") {
        Serial.print("! Tundmatu LED režiim, kasutan: kustus");
        Serial.println();
        LED_REZIIM = "kustus";
    }
    if (SYTTIB - KUSTUB < MIN_HYSTEREES) {
        Serial.print("! Hüsterees liiga kitsas (");
        Serial.print(SYTTIB - KUSTUB);
        Serial.print("), laiendan: kustub = ");
        KUSTUB = SYTTIB - MIN_HYSTEREES;
        Serial.println(KUSTUB);
    }
}

// =====
// EEPROM: konfig püsib üle voolukatkestuse
// =====
void salvestaKonfigEEPROM() {
    KonfigSalv s;
    s.magic = EEPROM_MAGIC;
    s.syttib = SYTTIB; s.kustub = KUSTUB; s.heledus = HELEDUS;
}

```

```

s.kuiv = KUIV; s.marg = MARG;
s.saada_min = SAADA_MIN; s.konfig_min = KONFIG_MIN;
s.temp_min = TEMP_MIN; s.temp_max = TEMP_MAX;
s.niiskus_min = NIISKUS_MIN; s.niiskus_max = NIISKUS_MAX;
s.muld_min = MULD_MIN; s.muld_max = MULD_MAX;
memset(s.led_reziim, 0, sizeof(s.led_reziim));
LED_REZIIM.toCharArray(s.led_reziim, sizeof(s.led_reziim));

KonfigSalv vana;
EEPROM.get(0, vana);
// kirjutatakse ainult muutuse korral (mälu kulumine)
if (memcmp(&vana, &s, sizeof(s)) != 0) {
    EEPROM.put(0, s);
    Serial.println(">> Konfig salvestatud EEPROM-i");
}
}

bool laeKonfigEEPROM() {
    KonfigSalv s;
    EEPROM.get(0, s);
    if (s.magic != EEPROM_MAGIC) return false;
    SYTTIB = s.syttib; KUSTUB = s.kustub; HELEDUS = s.heledus;
    KUIV = s.kuiv; MARG = s.marg;
    SAADA_MIN = max(1, s.saada_min);
    KONFIG_MIN = max(1, s.konfig_min);
    saada_vahe_ms = (unsigned long)SAADA_MIN * 60UL * 1000UL;
    konfig_vahe_ms = (unsigned long)KONFIG_MIN * 60UL * 1000UL;
    TEMP_MIN = s.temp_min; TEMP_MAX = s.temp_max;
    NIISKUS_MIN = s.niiskus_min; NIISKUS_MAX = s.niiskus_max;
    MULD_MIN = s.muld_min; MULD_MAX = s.muld_max;
    s.led_reziim[sizeof(s.led_reziim) - 1] = 0;
    LED_REZIIM = String(s.led_reziim);
    kontrolliKonfig();
    konfigLaetud = true;
    return true;
}

// =====
bool saadaAndmed(float t, float n, int m, int v, bool l) {
    String tee = String(WEBAPP_PATH) +
        "?action=saada" +
        "&id=" + String(SEADE_ID) +
        "&temp=" + String(t, 1) +
        "&niiskus=" + String(n, 0) +
        "&muld=" + String(m) +
        "&valgus=" + String(v) +
        "&led=" + (l ? "poleb" : "kustu");

    // aadressi (deployment-ID) jadaporti ei trükita
    Serial.println(">> Saadan andmed");

    String keha = "";
    if (!teeGetParing(HOST, tee, keha)) {
        Serial.println("X Saatmine ebaõnnestus");
        return false;
    }
    Serial.print(">> Vastus: "); Serial.println(keha);
    // rida on tabelis ainult siis, kui veebirakendus vastas ok:true
    if (keha.indexOf("\nok\:true") < 0) {
        Serial.println("X Tabel ei salvestanud rida");
        return false;
    }
    return true;
}

// =====
bool teeGetParing(const char* host, String tee, String& keha) {
    if (!klient.connect(host, 443)) {

```

```

    Serial.println("X HTTPS ühendus ebaõnnestus");
    return false;
}
klient.print("GET "); klient.print(tee);
klient.println(" HTTP/1.1");
klient.print("Host: "); klient.println(host);
klient.println("User-Agent: ArduinoR4WiFi");
klient.println("Connection: close");
klient.println();

String redirect = loeLocation();
klient.stop();
if (redirect.length() == 0) {
    Serial.println("X Redirect puudub");
    return false;
}
// Google suunab sisselogimislehele, kui juurdepääs ei ole Anyone
if (redirect.indexOf("accounts.google.com") >= 0) {
    Serial.println("X Sisselogimisleht: juurdepääs ei ole Anyone");
    return false;
}

String url    = redirect.substring(8);
int sl       = url.indexOf('/');
String host2 = url.substring(0, sl);
String tee2  = url.substring(sl);

if (!klient.connect(host2.c_str(), 443)) {
    Serial.println("X Teine ühendus ebaõnnestus");
    return false;
}
klient.print("GET "); klient.print(tee2);
klient.println(" HTTP/1.1");
klient.print("Host: "); klient.println(host2);
klient.println("User-Agent: ArduinoR4WiFi");
klient.println("Connection: close");
klient.println();

keha = loeKeha();
klient.stop();
return keha.length() > 0;
}

String loeLocation() {
    String loc = "";
    // veebirakendus vastab alles pärast salvestamist: ootus 20 s
    unsigned long deadline = millis() + 20000;
    while ((klient.connected() || klient.available())
        && millis() < deadline) {
        if (klient.available()) {
            String rida = klient.readStringUntil('\n');
            rida.trim();
            if (rida.length() == 0) break;
            if (rida.startsWith("Location: ")
                || rida.startsWith("location: ")) {
                loc = rida.substring(10);
                loc.trim();
            }
        }
    }
    return loc;
}

String loeKeha() {
    String keha = "";
    bool kehas = false;
    unsigned long deadline = millis() + 10000;
    while ((klient.connected() || klient.available())

```

```

        && millis() < deadline) {
    if (klient.available()) {
        String rida = klient.readStringUntil('\n');
        if (!kehas) { if (rida.length() <= 1) kehas = true; }
        // vastus on lühike; pikem tekst (nt HTML-leht) lõigatakse
        else if (keha.length() < 2000) keha += rida;
    }
}
return keha;
}

// =====
bool yhendaWifi() {
    // võrgu nime ega IP-aadressi jadaporti ei trükita
    Serial.println(">> WiFi ühendamine");
    WiFi.disconnect(); WiFi.end(); delay(1500);

    if (WiFi.status() == WL_NO_MODULE) {
        Serial.println("X WiFi moodul puudub");
        return false;
    }

    for (int katse = 1; katse <= 3; katse++) {
        Serial.print(">> Katse "); Serial.print(katse); Serial.print(": ");
        WiFi.begin(SSID, PAROOL);
        int n = 0;
        while (WiFi.status() != WL_CONNECTED && n < 30) {
            delay(500); Serial.print("."); n++;
        }
        Serial.println();

        if (WiFi.status() == WL_CONNECTED) {
            unsigned long ipDeadline = millis() + 5000;
            while (WiFi.localIP() == IPAddress(0, 0, 0, 0)
                && millis() < ipDeadline) delay(100);
            if (WiFi.localIP() == IPAddress(0, 0, 0, 0)) {
                Serial.println("X IP-aadress puudub, proovin uuesti");
                WiFi.disconnect(); WiFi.end(); delay(2000);
                continue;
            }
            Serial.println(">> WiFi OK, IP-aadress saadud");
            return true;
        }
        WiFi.disconnect(); WiFi.end(); delay(2000);
    }
    Serial.println("X WiFi ebaõnnestus");
    return false;
}

void vilgutaStatus(int n, int viiv) {
    for (int i = 0; i < n; i++) {
        digitalWrite(statusLed, HIGH); delay(viiv);
        digitalWrite(statusLed, LOW); delay(viiv);
    }
}

```

Lisa B – pilvetabeli veebirakendus Kood_naidis.gs

Kasutamine: peatükk 4, samm 2. Muuta tuleb ainult TABEL_ID. Funktsioonid seadistaTabel ja testiSalvestamist käivitatakse üks kord käsitsi (peatükk 4, sammud 3–4); seadistaTabel loob ka päästiku, mis käivitab funktsiooni arhiveeriAndmed iga kuu 1. päeval (peatükk 7). Ülejäänud töötab veebirakendusena: see võtab andmed vastu, peab lehte Seadmed ja annab seadmele seaded; haldusvaade ja e-posti teavitused ei kuulu sellesse versiooni.

```

// =====
// Terraariumi seire - Apps Scripti näidis (miinimum) v2.3
// Vastuvõtt: ?action=saada&id=..&temp=..&niiskus=..&muld=..
//           &valgus=..&led=..
// Konfig:   ?action=konfig&id=..
// Lehed: Andmed (logi), Seadmed (viimane olek),
//       Konfig (seaded seadme kohta)
// Arhiiv: iga kuu 1. päeval kell 1 kopeeritakse leht Andmed
//       eraldi faili Terraarium_arhiiv_AAAA_KK
// =====

// pilvetabeli aadressist: /spreadsheets/d/<ID>/edit
const TABEL_ID = "SIIA_TABELI_ID";

const ANDMETE_LEHT = "Andmed";
const SEADMETE_LEHT = "Seadmed";
const KONFIG_LEHT = "Konfig";

// syttib - kustub vähemalt 30; heledus 0-255
const VAIKIMISI = {
  syttib: 190, kustub: 150, heledus: 150,
  kuiv: 620, marg: 300,
  saada_min: 5, konfig_min: 5,
  temp_min: 10, temp_max: 35,
  niiskus_min: 30, niiskus_max: 100,
  muld_min: 15, muld_max: 90,
  led_reziim: "auto"
};
const KONFIG_VEERUD = ["ID", "syttib", "kustub", "heledus", "kuiv",
  "marg", "saada_min", "temp_min", "temp_max",
  "niiskus_min", "niiskus_max", "muld_min",
  "muld_max", "led_reziim", "konfig_min"];
const STRING_VALJAD = { led_reziim: true };
const LED_REZIIMID = ["auto", "sees", "kustus"];

// -----
function doGet(e) {
  var p = (e && e.parameter) ? e.parameter : {};
  if (p.action === "saada") return salvestaAndmed(parseAndmed(p));
  if (p.action === "konfig") {
    return JSONvastus({ ok: true, konfig: loeKonfig(p.id) });
  }
  return JSONvastus({ ok: false, error: "Tundmatu action" });
}

function JSONvastus(obj) {
  return ContentService.createTextOutput(JSON.stringify(obj))
    .setMimeType(ContentService.MimeType.JSON);
}

function parseAndmed(p) {
  return {
    id: p.id,
    temp: parseFloat(p.temp),
    niiskus: parseFloat(p.niiskus),
    muld: parseFloat(p.muld),
    valgus: parseInt(p.valgus, 10),
    led: p.led || ""
  };
}

// -----
function salvestaAndmed(d) {
  var lock = LockService.getScriptLock();
  try { lock.waitLock(10000); }
  catch (e) { return JSONvastus({ ok: false, error: "Lukk hõivatud" }); }
  try {
    var ss = SpreadsheetApp.openById(TABEL_ID);
  }
}

```

```

    var leht = ss.getSheetByName(ANDMETE_LEHT) || looAndmeteLeht(ss);
    var aeg = new Date();
    leht.appendRow([aeg, d.id, d.temp, d.niiskus, d.muld,
                    d.valgus, d.led]);
    uuendaSeadmeOlek(ss, d, aeg);
    tagaKonfigOlema(ss, d.id);
    lock.releaseLock();
    return JSONvastus({ ok: true, aeg: aeg.toISOString() });
} catch (err) {
    lock.releaseLock();
    return JSONvastus({ ok: false, error: err.message });
}
}

// staatus näitab, milline näit on piiridest väljas
function uuendaSeadmeOlek(ss, d, aeg) {
    var leht = ss.getSheetByName(SEADMETE_LEHT) || looSeadmeteLeht(ss);
    var andmed = leht.getDataRange().getValues();
    var rida = -1;
    for (var i = 1; i < andmed.length; i++) {
        if (andmed[i][0] === d.id) { rida = i + 1; break; }
    }
    var k = loeKonfig(d.id);
    var hoiatused = [];
    if (d.temp > k.temp_max || d.temp < k.temp_min) {
        hoiatused.push("temperatuur");
    }
    if (d.niiskus > k.niiskus_max || d.niiskus < k.niiskus_min) {
        hoiatused.push("õhuniiskus");
    }
    if (d.muld < k.muld_min) hoiatused.push("muld kuiv");
    if (d.muld > k.muld_max) hoiatused.push("muld märg");
    var staatus = hoiatused.length
        ? "HOIATUS: " + hoiatused.join(", ") : "OK";
    var uued = [d.temp, d.niiskus, d.muld, d.valgus, d.led, aeg, staatus];
    if (rida > 0) leht.getRange(rida, 3, 1, 7).setValues([uued]);
    else
        leht.appendRow([d.id, ""].concat(uued));
}

// -----
function loeKonfig(id) {
    if (!id) return Object.assign({}, VAIKIMISI);
    var ss = SpreadsheetApp.openById(TABEL_ID);
    var leht = ss.getSheetByName(KONFIG_LEHT);
    if (!leht) return Object.assign({}, VAIKIMISI);
    var andmed = leht.getDataRange().getValues();
    for (var i = 1; i < andmed.length; i++) {
        if (andmed[i][0] !== id) continue;
        var k = {};
        for (var j = 1; j < KONFIG_VEERUD.length; j++) {
            var nimi = KONFIG_VEERUD[j], v = andmed[i][j];
            if (STRING_VALJAD[nimi]) {
                var s = String(v === undefined ? "" : v).trim().toLowerCase();
                k[nimi] = s !== "" ? s : VAIKIMISI[nimi];
            } else {
                k[nimi] = (v !== "" && v !== undefined && !isNaN(v))
                    ? parseFloat(v) : VAIKIMISI[nimi];
            }
        }
    }
    return k;
}
return Object.assign({}, VAIKIMISI);
}

function tagaKonfigOlema(ss, id) {
    var leht = ss.getSheetByName(KONFIG_LEHT) || looKonfigLeht(ss);
    var andmed = leht.getDataRange().getValues();
    for (var i = 1; i < andmed.length; i++) {
        if (andmed[i][0] === id) return;
    }
}

```

```

}
var uus = [id];
for (var j = 1; j < KONFIG_VEERUD.length; j++) {
    uus.push(VAIKIMISI[KONFIG_VEERUD[j]]);
}
leht.appendRow(uus);
}

// -----
// lehtede loomine
function pais(leht, veerud) {
    leht.appendRow(veerud);
    leht.getRange(1, 1, 1, veerud.length).setFontWeight("bold")
        .setBackground("#1b4332").setFontColor("#fff");
    leht.setFrozenRows(1);
    return leht;
}
// uuel lehel on 26 veergu; üleliigsed kustutatakse, sest
// pilvetabeli 10 miljoni lahtri piiri arvestatakse kõigi veergudega
function looAndmeteLeht(ss) {
    var leht = pais(ss.insertSheet(ANDMETE_LEHT),
        ["Ajatempel", "Seadme ID", "Temperatuur °C", "Niiskus %",
        "Muld %", "Valgus", "LED"]);
    leht.deleteColumns(8, leht.getMaxColumns() - 7);
    return leht;
}
function looSeadmeteLeht(ss) {
    return pais(ss.insertSheet(SEADMETE_LEHT),
        ["Seadme ID", "Asukoht", "Temperatuur", "Niiskus", "Muld",
        "Valgus", "LED", "Viimane uuendus", "Staatus"]);
}
// veerus led_reziim on rippmenüü auto / sees / kustus
function lookonfigLeht(ss) {
    var leht = pais(ss.insertSheet(KONFIG_LEHT), KONFIG_VEERUD);
    var veerg = KONFIG_VEERUD.indexOf("led_reziim") + 1;
    var reegel = SpreadsheetApp.newDataValidation()
        .requireValueInList(LED_REZIIMID, true)
        .setAllowInvalid(false).build();
    leht.getRange(2, veerg, 200, 1).setDataValidation(reegel);
    return leht;
}

// Käivita üks kord käsitsi (Run -> seadistaTabel), et lehed
// tekiksid, õigused antaks ja kuarhiivi päästik loodaks.
function seadistaTabel() {
    var ss = SpreadsheetApp.openById(TABEL_ID);
    if (!ss.getSheetByName(ANDMETE_LEHT)) looAndmeteLeht(ss);
    if (!ss.getSheetByName(SEADMETE_LEHT)) looSeadmeteLeht(ss);
    if (!ss.getSheetByName(KONFIG_LEHT)) lookonfigLeht(ss);
    var olemas = ScriptApp.getProjectTriggers().some(function (t) {
        return t.getHandlerFunction() === "arhiveeriAndmed";
    });
    if (!olemas) {
        ScriptApp.newTrigger("arhiveeriAndmed").timeBased()
            .onMonthDay(1).atHour(1)
            .inTimezone(ss.getSpreadsheetTimezone()).create();
    }
    Logger.log("Tabel valmis: " + ss.getUrl());
}

// -----
// Kuuarhiiv: leht Andmed kopeeritakse uude faili (Minu Drive),
// kopeeritud read kustutatakse. Koopia ajal saabunud read jäävad
// lehele Andmed ega satu arhiivi kaks korda.
function arhiveeriAndmed() {
    var ss = SpreadsheetApp.openById(TABEL_ID);
    var leht = ss.getSheetByName(ANDMETE_LEHT);
    if (!leht) return;
    var n = leht.getLastRow() - 1;

```

```

if (n < 1) return;
var eile = new Date(Date.now() - 24 * 3600 * 1000);
var nimi = "Terraarium_arhiiv_" + Utilities.formatDate(
    eile, ss.getSpreadsheetTimeZone(), "yyyy_MM");
var arhiiv = SpreadsheetApp.create(nimi);
var koopia = leht.copyTo(arhiiv).setName(ANDMETE_LEHT);
arhiiv.deleteSheet(arhiiv.getSheets()[0]);
if (koopia.getLastRow() > n + 1) {
    koopia.deleteRows(n + 2, koopia.getLastRow() - n - 1);
}
// lukku ei võeta: seadmete saatmine jätkub ka kustutamise ajal
// üks tühi rida jääb alles: kõiki päiseta ridu ei saa kustutada
leht.insertRowsAfter(leht.getMaxRows(), 1);
leht.deleteRows(2, n);
}

// Katse ilma plaadita: Run -> testiSalvestamist,
// seejärel vaata lehte Andmed.
function testiSalvestamist() {
    salvestaAndmed({ id: "T00", temp: 21.5, niiskus: 55, muld: 40,
        valgus: 300, led: "kustu" });
}

```