

# IOT-projekti

Perjantaina 8.9.2023 (1. kerta)

IOT projektissa olisi idea luodaan IOT-laite. Käytössä on Arduino, Node Mcu ja vielä lisäksi Raspberry Pi. **Kurssi ja sen esimerkit keskittyvät Arduino IOT 33 Nanon ympärille.**

HUOM. Jos olet jo harrastanut jonkun näköistä Arduinon kanssa touhuamista tai vastaavaa, niin tämä kurssi ei välttämättä opeta sinulle mitään hirveästi uutta.

Tällä kurssilla tehdään hyvin yksinkertaisia harjoitteita. Mikä kurssista tekee jollakin tavalla haastavan, on että me emme keskity vain yhteen osaan vaan toteutamme laitteen ja palvelinpään ohjelmiston (ekosysteemi?). Kurssin esimerkki laite on lämpösensorilla varustettu Arduino, joka saa ohjeensa palvelimelta. Ohjeiden mukaan laite osaa päättää liikuttaako servomoottoria vai ei.

Arduino IOT 33 Nano. On jo itsessään WIFI-kykyinen laite. WIFI-ominaisuutta ei siis onneksi tarvitse käytössämme olevaan Nanoon lisätä.

Itsessään servon tai lämpöanturin liittäminen Arduinoon ei ole mikään kauhean hankala juttu. Vielä oikeastaan servon hallitseminen lämpödatallakaan ei ole hirveän haastavaa.

Tulemme hallitsemaan servomoottorin käynnistymistä ulkoisesta palvelusta, jonka me tällä kurssilla myös rakennamme. Palvelun data on tosi yksinkertainen minimilämpötila ja maksimilämpötila. Näiden kahden lämpötilan perusteella servomoottori liikkuu haluttuun asentoon. Saman voi testata ensin yksinkertaisemmin ledeillä. Siis punainen vihreä ledi kelpaavat aluksi aivan yhtä hyvin kunhan Arduino perustaa päätöksensä ulkoisiin parametreihin.

Palvelinsofta ei tule olemaan mikään kovin monimutkainen. Hyvä, jos sitä pystyy edes softaksi kutsumaan. Palvelu on oikeasti muutama rivi koodia PHP-kielillä. Te voitte itse päättää millä kielellä kirjoitatte palvelin softan, mutta tällä kurssilla se tullaan kirjoittamaan yksinkertaisesti PHPllä. Parametrit tullaan lukemaan ulkoisesta tiedostosta, joka on sitten ainakin meikäläisen esimerkissä niin .TXT muotoinen.

Palvelimen softan lisäksi tarvitaan hallintasivusto, josta parametreja voi sitten helposti muuttaa. Taas voitte itse määrittää tai päättää millä kielellä haluatte tämän kyseisen asian toteuttaa.

Projektimuotoisesti olette vapaita tekemään oman tyyllisen laitteen. Sama koskee tosiaan koodaamista. Koodatkaa vaikka basicilla, kuitenkin esimerkit tehdään Aduino IDEssä ja pieni pätkä PHPlla. Koulu tarjoaa tietyn määrän laitteita kurssille eli on tosiaan kuten mainittuja niin on Node MCU, Arduino ja Raspberry Pi. Voit itse päättää mitä laitetta haluat käyttää tähän projektiin.

Kurssi ei tule käsittelemään valitettavasti tietoturvaa eli laitteet mitä toteutatte saattaa vuotaa nettiin ihan mitä vaan. Kurssin ideana on lähinnä toteuttaa laite, joka kommunikoi/lukee dataa ulkoisesta palvelusta ja tekee päätöksiä sen mukaan mitä se sieltä palvelimelta saa.

Tekniikan labrasta löytyy lisää komponentteja lainaan, jos haluatte lisä maustaa laitettanne (pyydä opettajilta). Voitte myös suunnitella ja toteuttaa laitteellenne oman 3D-printatun kotelon halutessanne.

Kurssi etenee kuten Let's Make Game-kurssi. Projekti toteutetaan ryhmissä, ryhmän koko on 3–4 henkilöä (**viiden ryhmä on liian iso, koska kaikille ei riitä tarpeeksi tehtävää**).

Voitte toteuttaa projektin myös itsenäisesti, riippuu ihan miten paljon laitteita jää jaettavaksi.

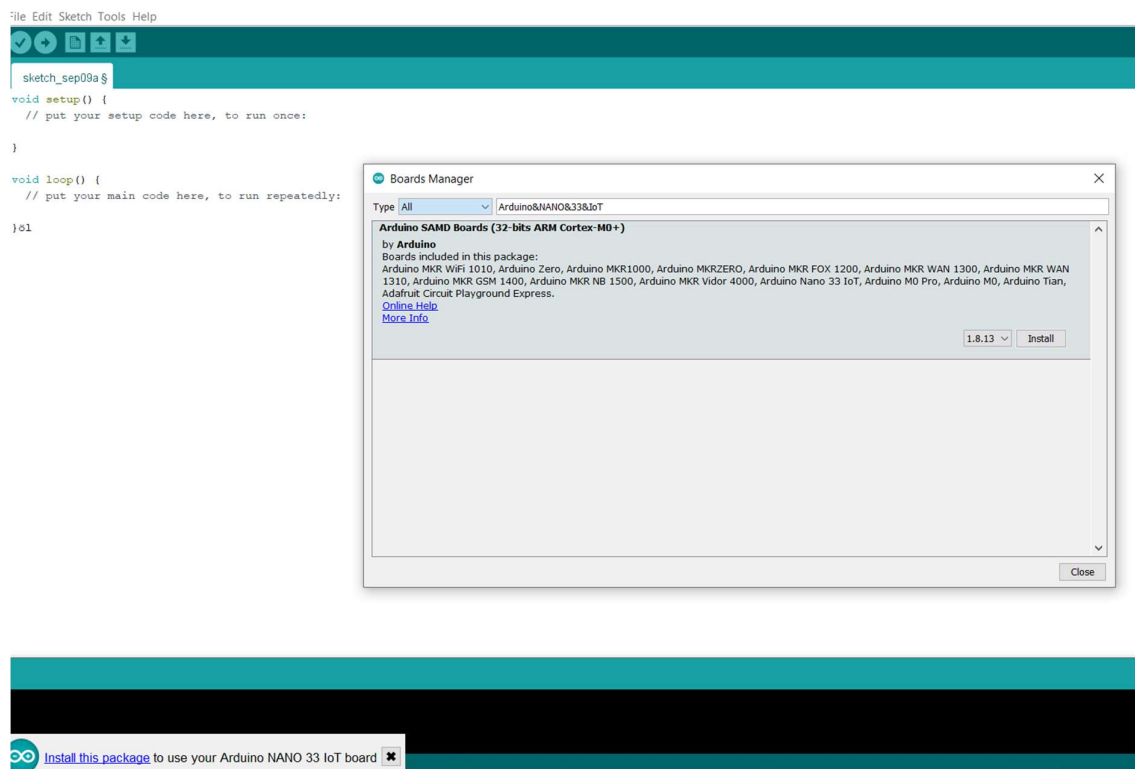
**Mikäli haluatte toteuttaa jotain muuta** mitä kurssilla ohjatusti tehdään, niin te luotte siitä projektisuunnitelman ja hyväksytätte sen. Suunnitelmassa on listattuna tietenkin myös komponentit mitä te tulette tarvitsemaan tuohon projektiin ne ja tietysti onko ne saatavilla. Projektiahan ei kannata lähteä toteuttamaan, jos teillä ei ole saatavilla tarvittavia komponentteja.

Jotta kurssi olisi oikea IOT kurssi niin pyritään myös tekemään verkottuvat laitteet eli niin, että toinen laite osaisi kommunikoida toisen laitteen kanssa ja tehdä päätöksiä sen mukaan. Tämä jää luultavasti tällä kurssilla vain teoria tasolle / jatkokehitysvaiheeseen.

Yhteistä aikaa kurssilla on hyvin vähän.

Aloitetaan

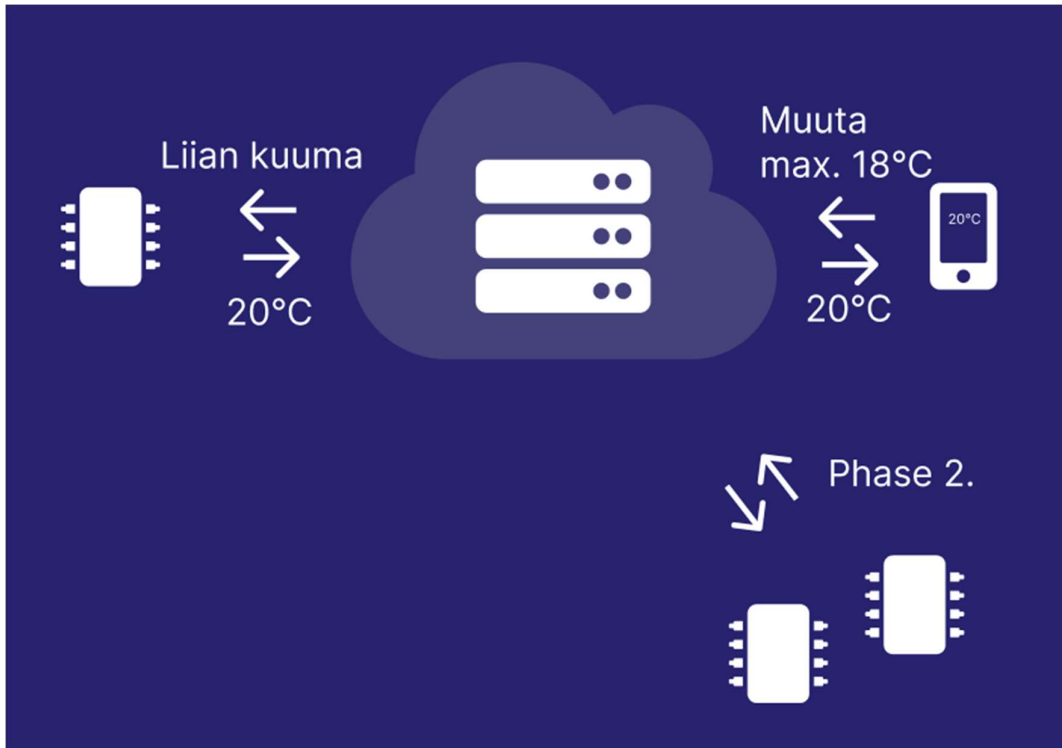
- alkutoimet
- yksinkertainen ohjelma serial monitoriin
- projektin anto, ryhmiin jako
- kotitehtävä: suunnitelma ja tarpeet



Mitä kurssilla voi tehdä / millaisia laitteita materiaaleista saa rakennettua?

Kurssilla toteutatte:

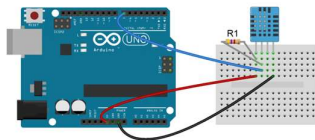
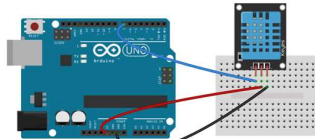
Lämpötilaa tarkkailevan (verkottuva) ja sen mukaan päätöksiä tekevän laitteen (ilmaräppänä).



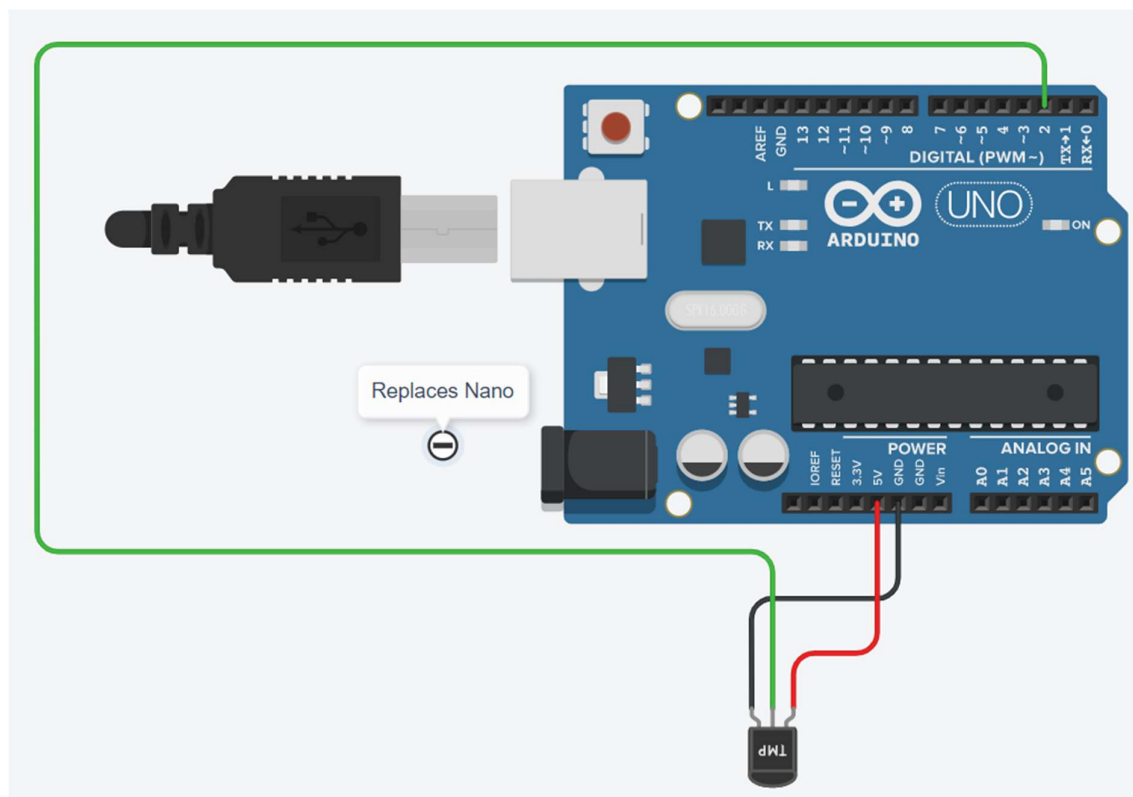
Virtuaalisesti Tinkercadilla paljon muutakin (ei kelpaa kurssin päättötyöksi)

Myös tätä voi koittaa <https://wokwi.com/> (ei kelpaa kurssin päättötyöksi)

- suunnitelman esittely
- projektin aloitus
- johdotus (lämpöanturi) (3.3v pinni riittää)
- kotitehtävä / tunnilla: lämpötilan lukeminen
- lisätehtävä: jos lämpötila on yli 24 celsiusta sano serial-monitoriin "Kuuma".  
Jos lämpötila välillä 22-24 sano serial-monitoriin "Just hyvä".  
Jos lämpötila jotain muuta sano serial-monitoriin "Kylmä"



Lähde: <https://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-an-arduino/>



Perjantaina 22.9.2022 (3. kerta)

- lämpötilan lukemistehtävän esittely
- palvelinpuolen rakennus (php, node js, python ...)
- tehtävä / kotitehtävä: tiedon lukeminen palvelimelta

Tulen näyttämään pari esimerkkiä backendistä. Toinen PHP ja toinen Node Red. Käytettävä tekniikka on se ja sama, kunhan lopputulos on vasta pyydettyä/suunnitelmaa.

Perjantaina 29.9.2022

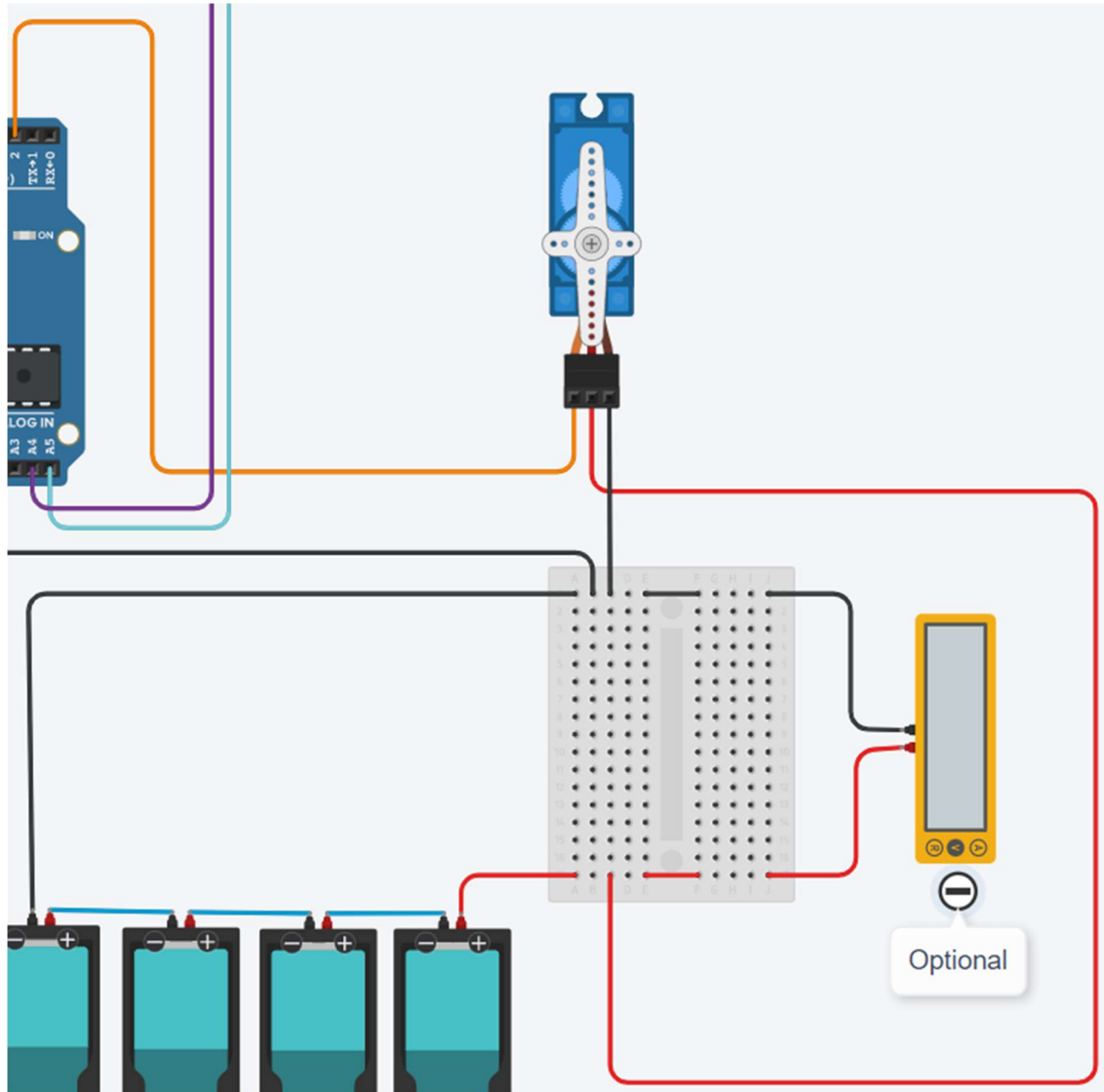
-tiedon lukeminen palvelimelta tehtävän esittely

-tehtävä / kotitehtävä: lämpötilan vertaaminen palvelimen muuttuviin arvoihin

Perjantaina 6.10.2022

- servon lisääminen yhtälöön + erillinen virtalähde (4x aa)
- tehtävä / kotitehtävä: servon hallinta palvelimen lämpötilojen mukaan
- servon voi myös halutessaan korvata ledillä (pyydä opettajalta)

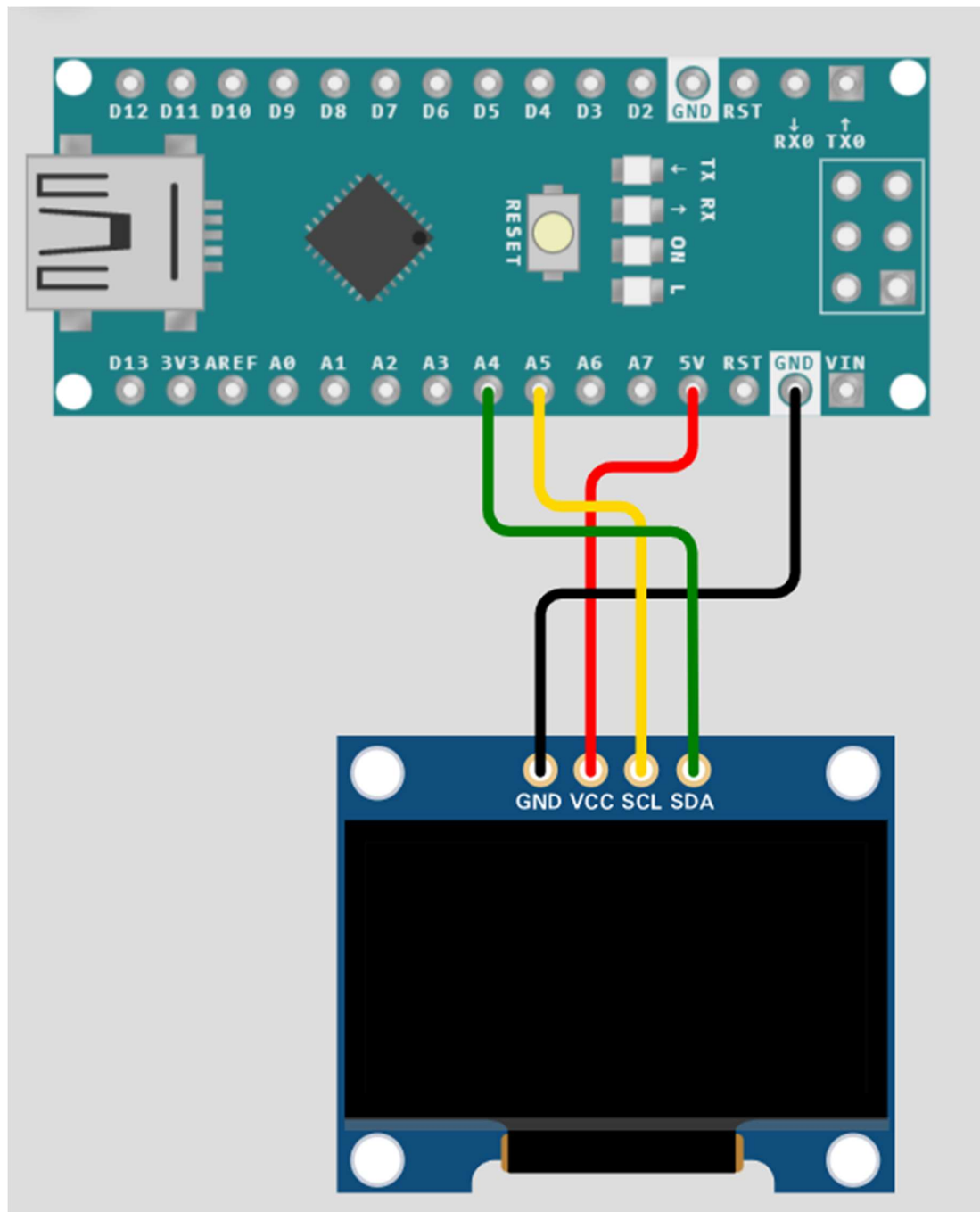
ÄLÄ kärväytä mitään! Tarkistetaan liitännät yhdessä.



Perjantaina 13.10.2022

-näytön lisääminen yhtälöön (vaatii/riittää 3.3v)

-tehtävä / kotitehtävä: lämpötilan näyttäminen näytöllä





Perjantaina 20.10.2022

-kaverin laitteen kanssa kommunikointi?

-tehtävä: Laitteen esittely ryhmissä 5-10min

## Sisältösuunnitelma

Perjantaina 9.9.2022

- alkutoimet
- yksinkertainen ohjelma serial monitoriin
- projektin anto, ryhmiin jako
- kotitehtävä: suunnitelma ja tarpeet

Perjantaina 16.9.2022

- suunnitelman esittely (ryhmä 1)
- projektin aloitus
- johdotus (lämpöanturi)
- kotitehtävä: lämpötilan lukeminen

Perjantaina 23.9.2022

- lämpötilan lukemistehtävän esittely
- palvelinpuolen rakennus (php)
- kotitehtävä: tiedon lukeminen palvelimelta

Perjantaina 30.9.2022

- tiedon lukeminen palvelimelta tehtävän esittely
- tehtävä / kotitehtävä: lämpötilan vertaaminen palvelimen muuttuviin arvoihin

Perjantaina 7.10.2022

- servon lisääminen yhtälöön + erillinen virtalähde (4x aa)
- tehtävä / kotitehtävä: servon hallinta palvelimen lämpötilojen mukaan

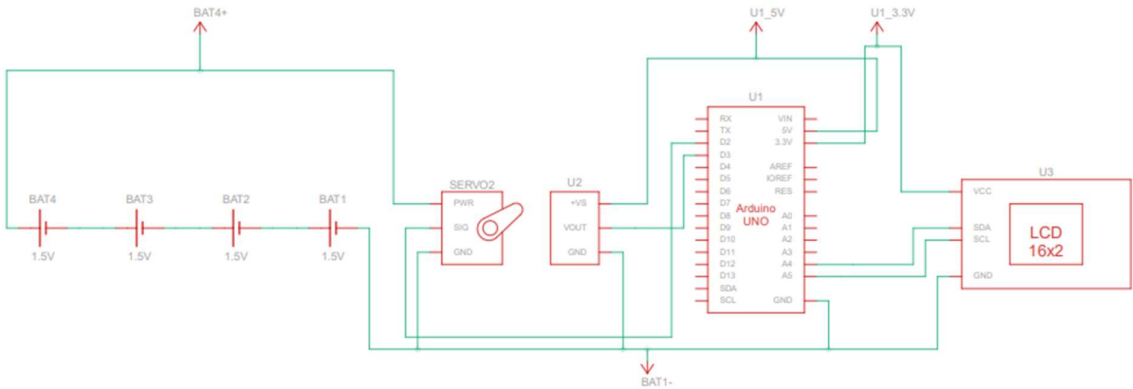
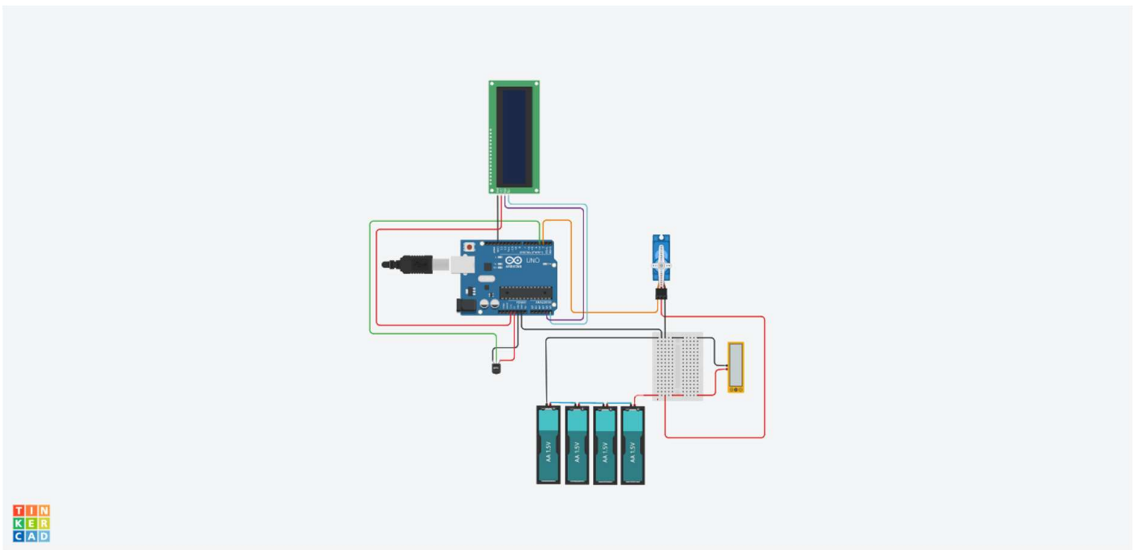
Perjantaina 14.10.2022

- näytön lisääminen yhtälöön
- tehtävä / kotitehtävä: lämpötilan näyttäminen näytöllä

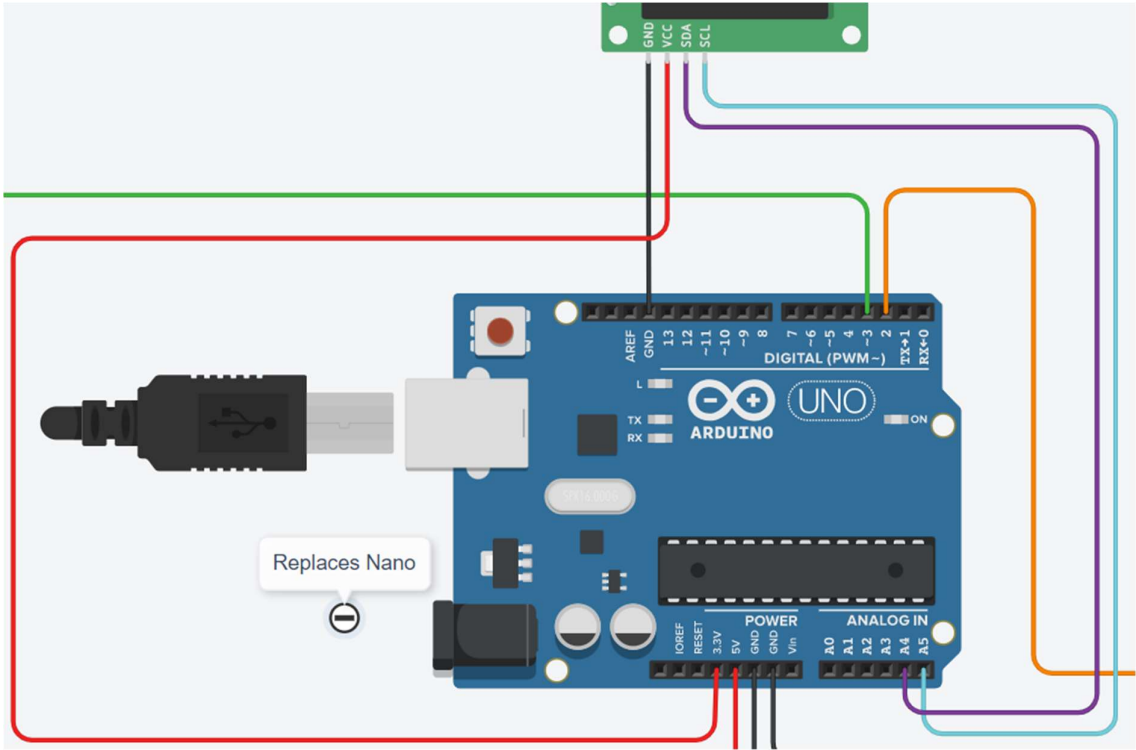
Perjantaina 21.10.2022

- kaverin laitteen kanssa kommunikointi?
- tehtävä: Laitteen esittely ryhmissä 5-10min

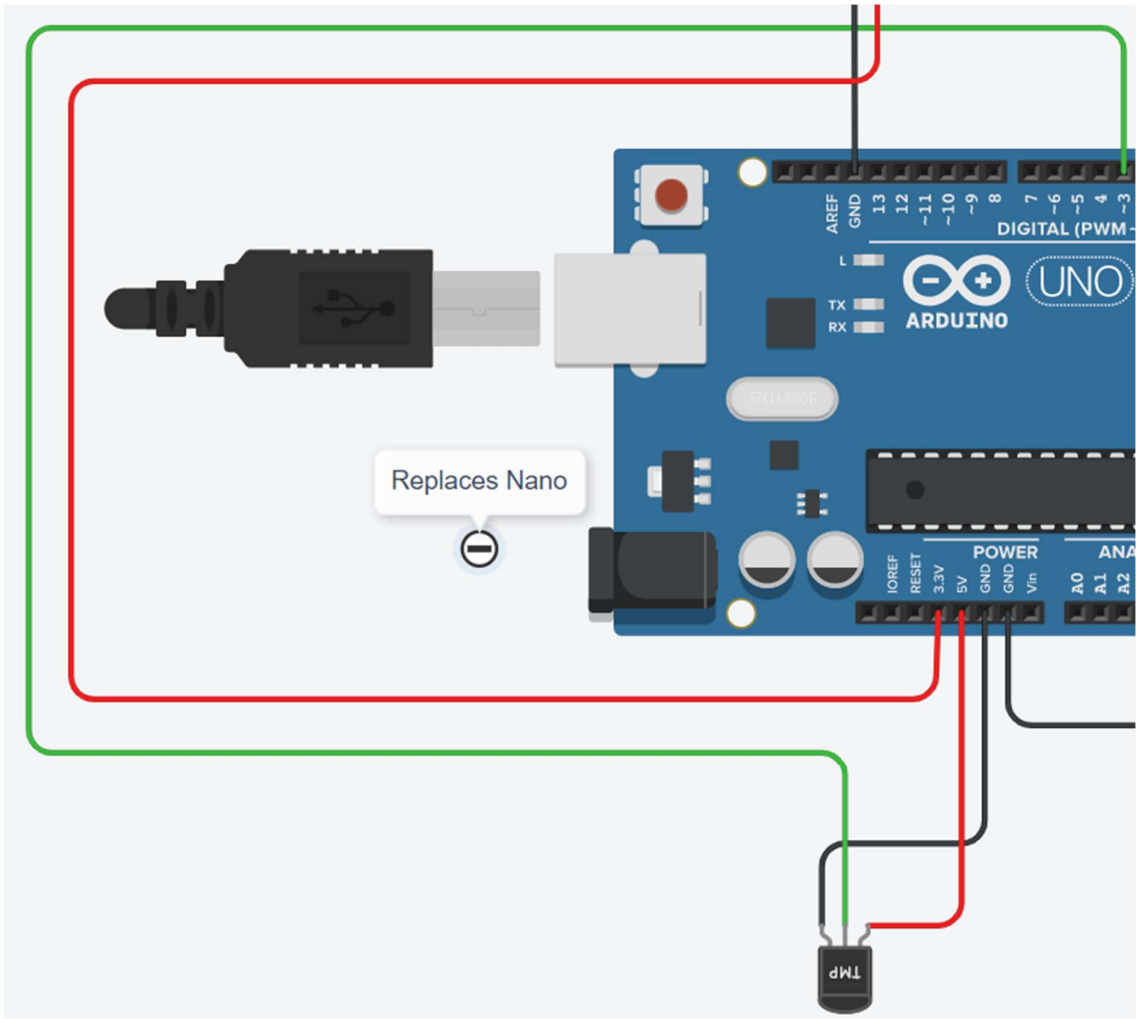
Lopullinen tuote



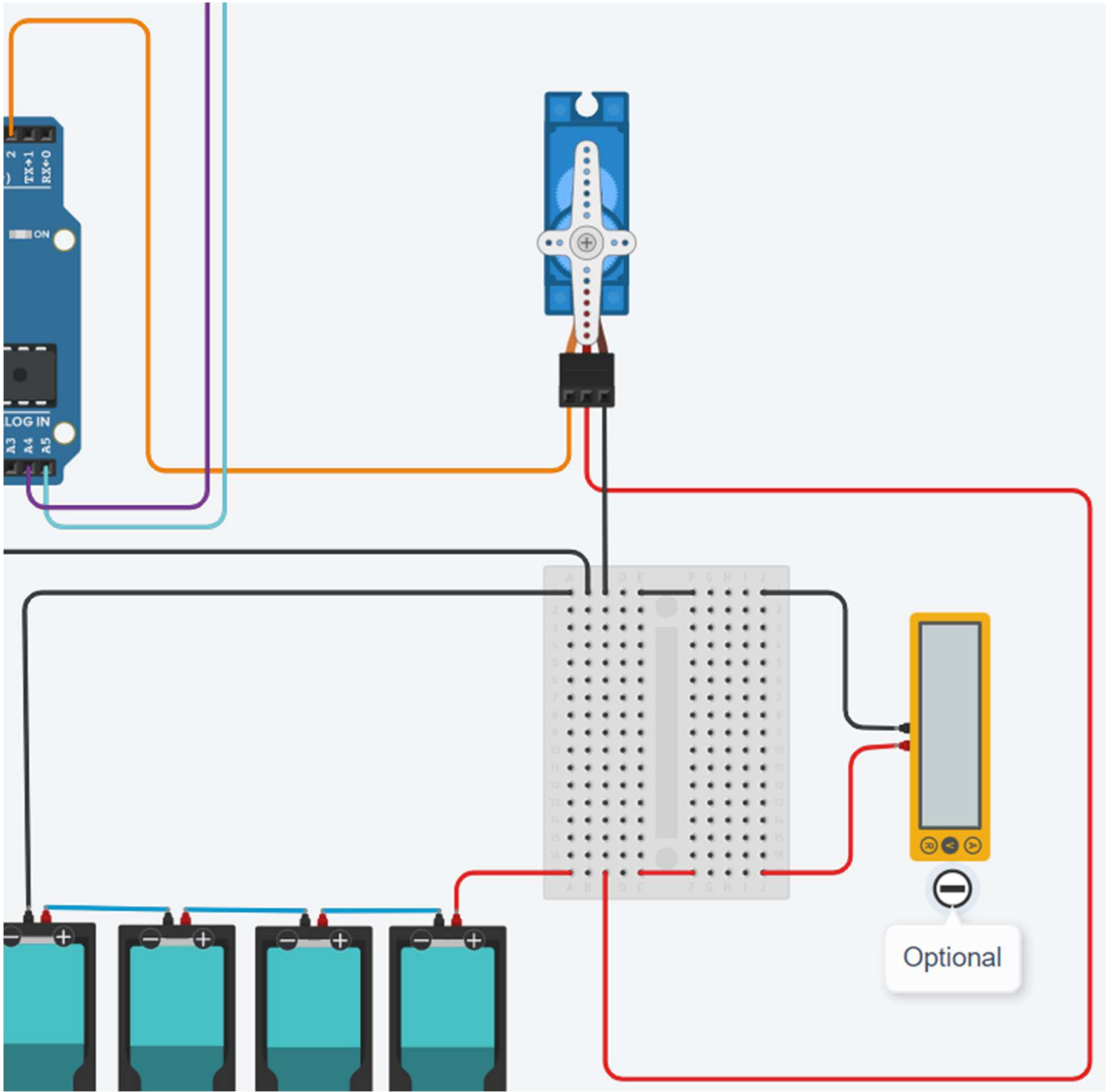
Näyttö



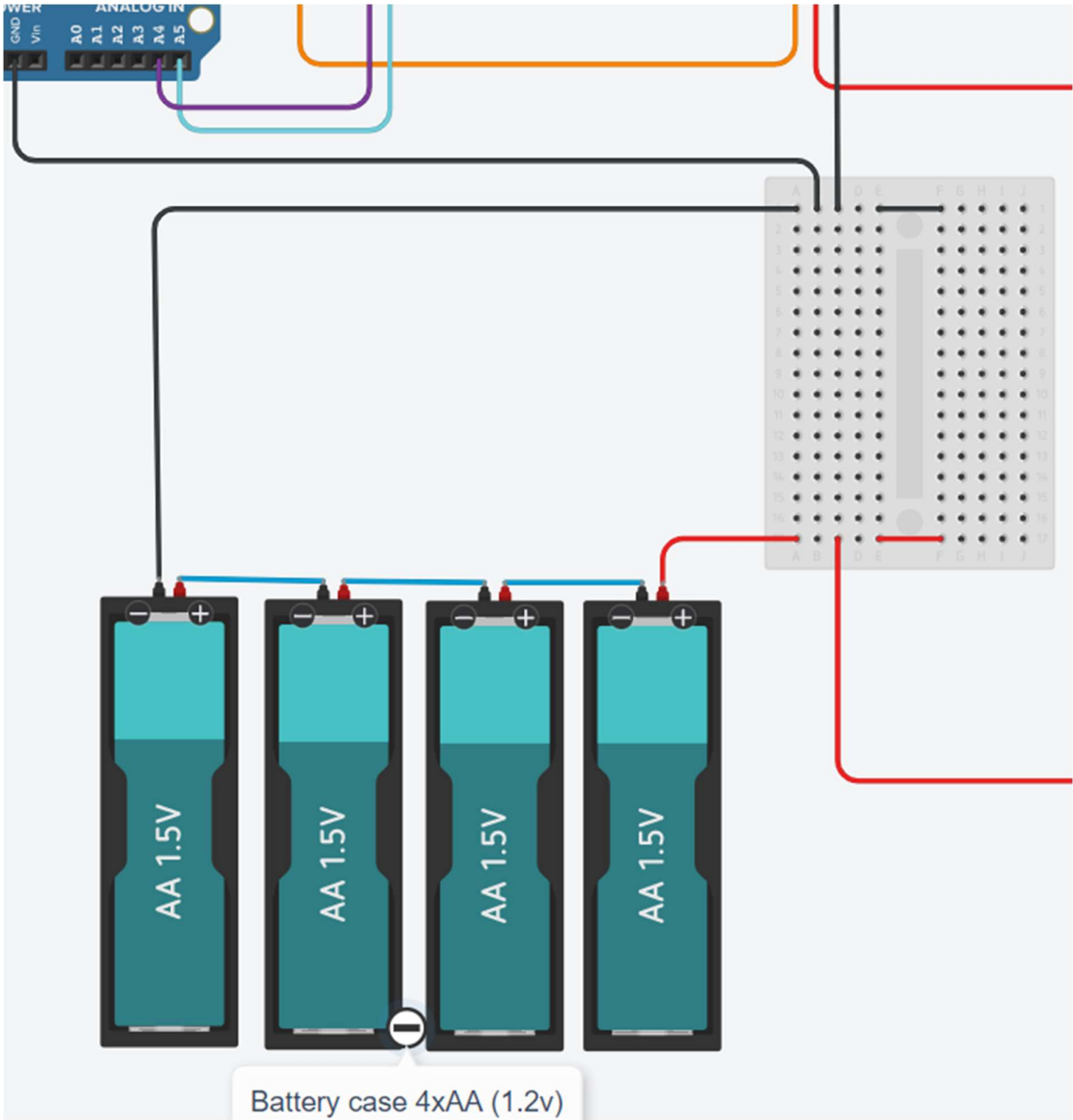
Lämpösensori



Servo



Virtalähde



## IFTTT

```
<?php
//https://stackoverflow.com/questions/7564832/how-to-bypass-access-
control-allow-origin
//header('Access-Control-Allow-Origin: *');
header('Access-Control-Allow-Origin: https://getsome.fi');
header("Content-Type: application/json; charset=UTF-8");

$threshold = 10;

function ifttt_send($heater_on,$temp,$humi){
    //http://www.giuseppeparrello.it/en/net_domotics_send_ifttt_event.php
    $apikey = "qwe"; //https://ifttt.com/maker_webhooks/settings
    $event = "heating_off";
    if($heater_on){
        $event = "heating_on";
    }

    $value3 = "From somewhere";

    $ch = curl_init();

    $postdata = json_encode([
        "temperature" => $temp,
        "humidity" => $humi,
        "value3" => $value3,
    ]);

    $header = array();
    $header[] = "Content-Type: application/json";

    curl_setopt($ch,CURLOPT_URL,
"https://maker.ifttt.com/trigger/$event/json/with/key/$apikey");
    curl_setopt($ch,CURLOPT_HTTPHEADER, $header);
    curl_setopt($ch,CURLOPT_POST, 1);
    curl_setopt($ch,CURLOPT_POSTFIELDS, $postdata);
    curl_setopt($ch,CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch,CURLOPT_SSL_VERIFYPEER, false);

    $result = curl_exec($ch);

    print_r($result);
    print_r("\r\n\r\n");

    curl_close($ch);
}

function makeConn(){
```



```

    $servername = "localhost";
    $username = "asd";
    $password = "dfg";
    $dbname = "ghj";

    $conn = new mysqli($servername, $username, $password, $dbname);

    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    return $conn;
}

function saveData($t,$h){
    $conn = makeConn();

    $sql = "INSERT INTO weatherdata (temperature, humidity, data_index,
created) VALUES ($t, $h, NULL, CURRENT_TIMESTAMP)";

    $response = "";
    if ($conn->query($sql)) {
        $response = "Record inserted successfully.";
    }
    if ($conn->errno) {
        $response = "Could not insert record into table: %s" . $conn-
>error . " kysely: " . $sql;
    }

    //ifttt stuff
    if(intval($t) < $threshold){
        ifttt_send(true,$t,$h);
    }else{
        ifttt_send(false,$t,$h);
    }

    return $response;

    $conn->close();
}

if( isset($_GET['temp']) && isset($_GET['humi']) ){
    $t = $_GET['temp'];
    $h = $_GET['humi'];
    echo saveData($t, $h);
}else{
    if( isset($_GET['amount']) ){
        $conn = makeConn();
    }
}

```

```

        $sql = "SELECT * FROM weatherdata ORDER BY created DESC LIMIT " .
$_GET['amount'];

        $result = $conn->query($sql);
        $rows = array();
        if ($result->num_rows > 0) {
            while($row = $result->fetch_assoc()) {
                $rows[] = $row;
            }
        } else {
            $rows[] = "error";
        }
        print json_encode($rows);
    }else{
        $conn = makeConn();
        $sql = "SELECT * FROM weatherdata ORDER BY created DESC LIMIT 50";

        $result = $conn->query($sql);

        $rows = array();
        if ($result->num_rows > 0) {
            while($row = $result->fetch_assoc()) {
                $rows[] = $row;
            }
        } else {
            $rows[] = "error "; //. $conn->error;
        }

        print json_encode($rows);
    }
}
?>

```



# "heating\_off", Mökki TUPA

[Edit title](#)

by peytecom

**Connected**



Connected Sep 08, 2022

Last activity Sep 08, 2022

Run 440 times

Notify me when this runs



Check the log of your  
Applet runs

View activity

Realtime Applets usually  
run within 10 seconds

Check now

# "heating\_off", Mökki TUPA Activity



## Applet ran

Sep 08 - 3:37 PM



### "heating\_off", Mökki TUPA

The event named "heating\_off" occurred on the Maker Webhooks service



#### Webhooks

Receive a web request with a JSON payload

○ Expand for details



#### TP-Link Kasa

Turn off

○ Expand for details

