

Puutarhan seurantajärjestelmä

Järjestelmän kuvaus

Järjestelmällä seurataan kasvimaan ja kasvihuoneen maan sekä ilman lämpötila- ja kosteustietoja. Järjestelmällä voidaan myös käynnistää pumppu kastelua varten, sekä ohjata kastelu haluttuun kohteeseen. Järjestelmään kuuluu ilman lämpötilaa ja kosteutta seuraavia antureita sekä maan kosteutta ja lämpötilaa seuraavia antureita. Lisäksi kasteluyksikköön kuuluu päällekytkentärele sekä releet kastelun ohjaamiseen haluttuun kohteeseen. Anturit on kytketty NodeMcu -yksikköihin. Tiedot lähetetään pilvipalvelimelle ja keskusyksikkönä toimivalle raspberry Pi 3 yksikölle. Järjestelmä sijaitsee n. 50 kilometrin etäisyydellä kodista, joten järjestelmän toimintaa pitää voida tarkkailla ja huoltaa etäyhteyden kautta. Sitä varten tarvitaan nopea verkkoyhteys, joka toteutetaan 4G verkkoyhteydellä. Projektin ensimmäisessä vaiheessa toteutetaan ympäristötiedon mittaus ja tietojen lähettäminen pilvipalveluun. Myöhemmin toteutettavassa toisessa vaiheessa automatisoidaan kastelu ja kasvihuoneen ilmanvaihtoon liittyvät toiminnot.

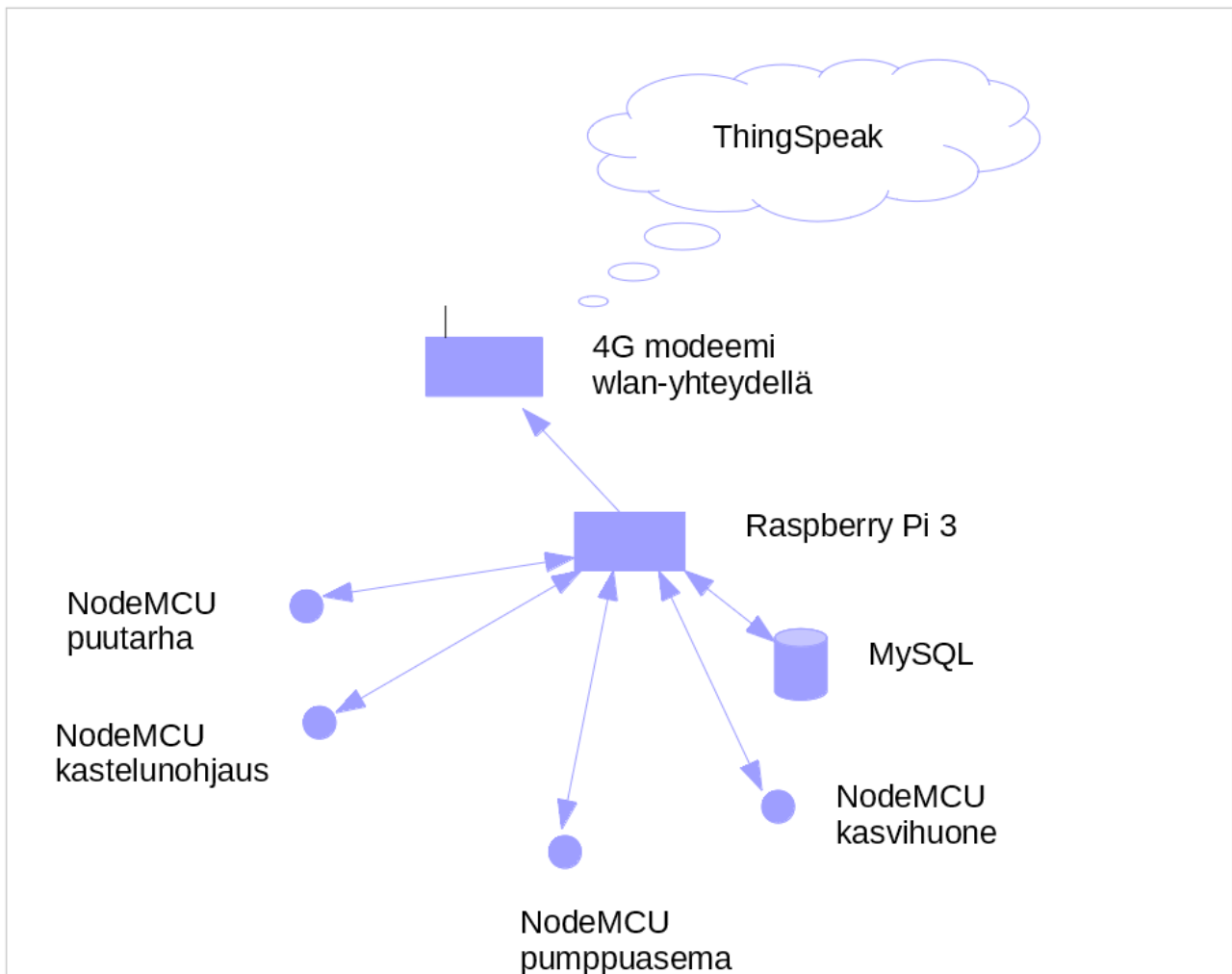
Järjestelmän mitaamat ympäristötiedot löytyvät ThingSpeakistä.

Linkki: <https://thingspeak.com/channels/268204>

Käytetyt komponentit

- 1 4G langaton reititin ja wlan yksikkö
- 1 raspberry pi -yksikkö tiedonkeruuseen ja tiedonvälitykseen.
- 3 NodeMcu -yksikköä mittaustiedon keruuta ja toimintojen ohjausta varten.
- 2 maan kosteutta mittaavaa anturia (FC-28-D).
- 3 maan lämpötilaa mittaavaa anturia (DIY DS18B20).
- 2 ilman lämpötilaa ja kosteutta mittaavaa anturia (dht11).
- 1 ultraäänianturi mittaamaan vedenpinnan korkeutta joessa (Hc -Sr04) (optio)
- 1 vedenpaineen mitta-anturi kasteluletkuun.
- 1 ohjattava kytkentärele pumpun käynnistämiseen.
- 1 ohjattava kytkentärele kastelun ohjaamiseen kohteeseen.
- (1-3) akkua anteteiden virtalähteeksi.
- 1 liitäntäjohtosetti. Universaali uros-naaras, uros-uros, naaras-naaras

Järjestelmäkaavio



Ohjelmistokuvaus

NodeMcu-yksiköt

Puutarha:

Kasvimaalla oleva NodeMcu -yksikkö mittaa maan kosteuden, maan lämpötilan, lämpötilan 20 cm:n korkeudella maasta sekä lämpötilan ja ilman kosteuden 1,5 m:n korkeudella maasta. Tiedot luetaan ja lähetetään Raspberry Pi 3:lle jatkokäsittelyä ja kasteluun liittyviä automatiotoimintoja varten mqtt viestinä puolen tunnin välein. Tiedot lähetetään myös suoraan ThingSpeakiin, josta niitä voi helposti seurata.

Kasvihuone:

Kasvihuoneessa oleva Node Mcu -yksikkö mittaa maan kosteuden ja ilman lämpötilan ja lähettää tiedot Raspberry Pi 3:lle mqtt viestinä jatkokäsittelyä ja kastelun ja ilmanvaihdon

automatisointia varten. Tiedot lähetetään myös suoraan ThingSpeakiin, josta niitä on helppo seurata.

Kastelupumpun kontrolloyksikkö:

Pumpun etäkäynnistystä varten kontrolloyksikössä on ohjattava rele. Koska jokiveden pinta voi laskea niin alas, että vettä ei ole, on pumpun käynnistymisen kuivana voitava estää. Tätä varten kontrolloyksikkö mittaa vedenpinnan korkeuden ja estää pumpun käynnistymisen, jos vedenpinta on asetetun raja-arvon alapuolella. Veden pinnan korkeustieto lähetetään myös ThingSpeakiin, josta sitä on helppo seurata.

Kastelun ohjausyksikkö:

Ohjausyksikössä on releohjaus, jolla kastelu voidaan ohjata haluttuun kohteeseen.

Raspberry Pi 3

Raspberry Pi 3:n tehtävänä on ohjata automaatioon liittyviä toimintoja, kuten kastelua ja kasvihuoneen ilmanvaihtoa. Tietoa voidaan myös kerätä tarpeen mukaan enemmän, kuin pivipalveluun lähetetään. Tässä on kyse lähinnä pilvipalvelun kustannuksista, eli paljonko dataa voi lähettää ilman kustannuksia.

NodeMcu -kykentäkaaviot

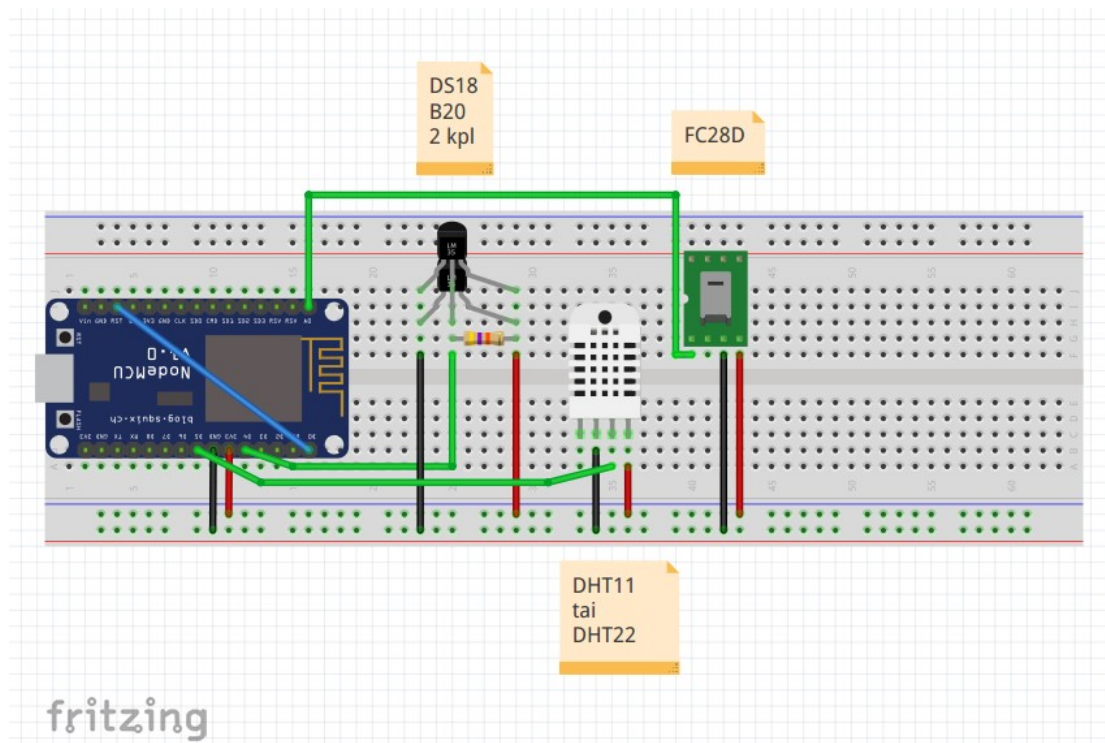
Lämpötilan ja kosteudenmittausyksikköjen kytkennät

Maan kosteudenmittausyksikön sensori A0 kytketään NodeMCU:n liitimeen A0.

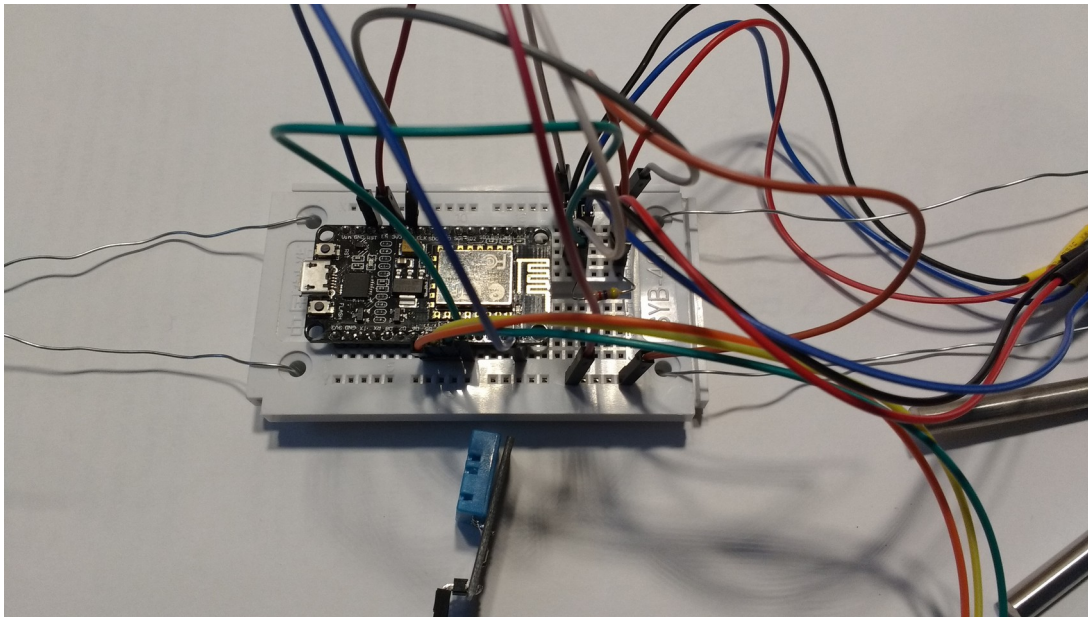
Lämpötilanmittausyksiköiden out-liitin kytketään NodeMCU:n digitaaliseen liitimeen D4.

Lämpötilan- ja kosteudenmittausyksikön out liitin kytketään NodeMCU:n digitaaliseen liitimeen D5.

Deep Sleep -moden herätystoimintoa varten D0 ja reset liittimet yhdistetään.



Kuva 1: Kaavio kytkennöistä kytkentälevyllä.



Kuva 2: Testiasennus koekytkentälevyllä



Kuva 3: Sääasema kasvimaalla

Node Mcu koodi lämpötila- ja kosteustietojen lukemiseen

NodeMCU-yksikkö tarvitsee kirjastot DHT.h, OneWire.h ja DallasTemperature lämpötilatietojen lukemiseen. Lukuväli on 15 minuuttia (900 sekuntia). DHT-sensori ilmoittaa lämpötilan ja ilmankosteuden, OneWire-sensorit ovat kosteita olosuhteita kestäviä. Nillä luetaan lämpötila maan pinnasta ja maasta. Koodi löytyy kokonaisuudessaan tiedostosta *sketch_may23a_mqtt_deep_sleep_omaserveri.ino*.

```
#include <PubSubClient.h>
#include <DHT.h>
#include <ESP8266WiFi.h>
#include <OneWire.h>
#include <WiFiClient.h>
#include <DallasTemperature.h>
#include <Streaming.h>

// DHT Sensor connected to digital pin 5
#define DHTPIN D5
// Type of DHT sensor
#define DHTTYPE DHT22
// Deep sleep delay
#define SLEEP_DELAY_IN_SECONDS 900 // Delay 15 minutes
// Waterproof temperature sensor
#define ONE_WIRE_BUS D4 // DS18B20 pin
```

```

// Initialize DHT sensor
DHT dht(DHTPIN, DHTTYPE);

// network SSID (name)
char ssid[] = "<network ssid>";
// network password
char pass[] = "<password>";

// Initialize the Wifi client library
WiFiClient client;
// Initialize the PubSubClient library
PubSubClient mqttClient(client);
// Define the MQTT broker
const char* server = "<server ip>";
const char* mqtt_username = "<mqtt username>";
const char* mqtt_password = "<mqtt password>";
const char* mqtt_topic = "<topic>";

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature DS18B20(&oneWire);

```

Aluksi alustetaan verkko ja mqtt-yhteydet.

```

void setup() {
  // Begin serial transfer
  Serial.begin(115200);
  dht.begin();
  delay(10);
  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  // Print the IP address
  Serial.println(WiFi.localIP());
  // Set the MQTT broker details
  mqttClient.setServer(server, 1883);
}

```

Lämpötilatiedot luetaan OneWire-sensoreista indeksin mukaan 0 ja 1

```

// Read temperature 0.2 m above ground
float getTemperature() {
  Serial << "Requesting DS18B20 temperature..." << endl;
  float temp;
  do {
    DS18B20.requestTemperatures();
    temp = DS18B20.getTempCByIndex(0);
    delay(100);
  } while (temp == 85.0 || temp == (-127.0));
}

```

```

    return temp;
}

```

Jatkuvasti toistettavassa osassa katkaistaan toiminnot väliajaksi ja yhdistetään uudelleen kun sleep-moodista herätään.

```

void loop() {
    // Check if MQTT client has connected else reconnect
    if (!mqttClient.connected())
    {
        reconnect();
    }
    // Call the loop continuously to establish connection to the server
    mqttClient.loop();

    mqttpublish();

    delay(100);

    Serial << "Closing MQTT connection..." << endl;
    mqttClient.disconnect();

    Serial << "Closing WiFi connection..." << endl;
    WiFi.disconnect();

    delay(100);

    Serial << "Entering deep sleep mode for " << SLEEP_DELAY_IN_SECONDS << "    seconds..." << endl;
    ESP.deepSleep(SLEEP_DELAY_IN_SECONDS * 1000000, WAKE_RF_DEFAULT);

    delay(500); // wait for deep sleep to happen
}

```

Publish-funktiossa suoritetaan tietojen luku ja lähetys mqtt-brokerille.

```

void mqttpublish() {
    // Read temperature from DHT sensor, Fahrenheit if dht.readTemperature(true)
    float t = dht.readTemperature();
    // Cut data length of all values. Last values in string are not otherwise accepted.
    String t_s = String(t, DEC);
    t_s = t_s.substring(0,7);

    // Read humidity from DHT sensor
    float h = dht.readHumidity();
    String h_s = String(h, DEC);
    h_s = h_s.substring(0,7);

    // Read from waterproof one wire temperature sensors
    // Read temperature 0.2 m above ground
    float t2 = getTemperature();
    String t2_s = String(t2, DEC);
    t2_s = t2_s.substring(0,7);
    // Read soil temperature
    float t3 = getTemperature2();
    String t3_s = String(t3, DEC);
    t3_s = t3_s.substring(0,7);

    // Create data string to send to broker

```



```
String data = String("field1=" + t_s + "&field2=" + h_s + "&field3=" + t2_s + "&field4=" + t3_s);  
// Get the data string length  
int length = data.length();  
char msgBuffer[length];  
// Convert data string to character buffer  
data.toCharArray(msgBuffer,length+1);  
Serial.println(msgBuffer);  
  
// Publish data to local broker.  
mqttClient.publish(mqtt_topic,msgBuffer);  
delay(500);  
}
```

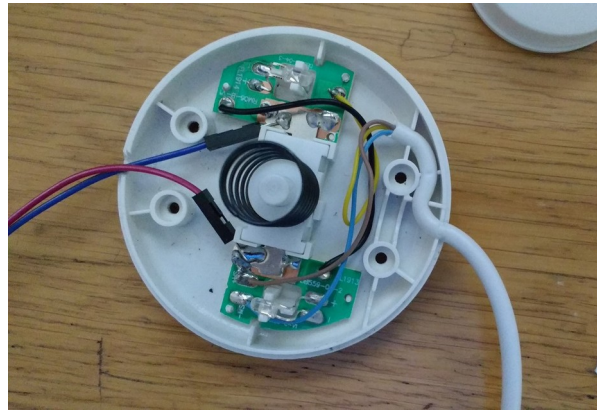
MQTT-serveri ja tietokantatoiminnot

MQTT-serverinä toimii Mosquitto, joka on asennettu Raspberry Pi 3:een. Serverikoodi, joka lukee tiedot brokerilta ja lähettää ne ThingSpeakiin, on tehty Python 3:lla.

mqtt_get_broker_data_main.py lukee tiedot brokerilta, *mqtt_save_broker_values_to_db.py* tallentaa arvot MySQL-kantaan ja *mqtt_send_garden_data.py* lähettää tiedot ThingSpeakiin. Koodit löytyvät kokonaisuudessaan näistä tiedostoista. *Mqtt_get_broker_data_main.py*-ohjelma käynnistyy automaattisesti, kun tietokone käynnistyy ja toimii taustalla omassa säikeessään.

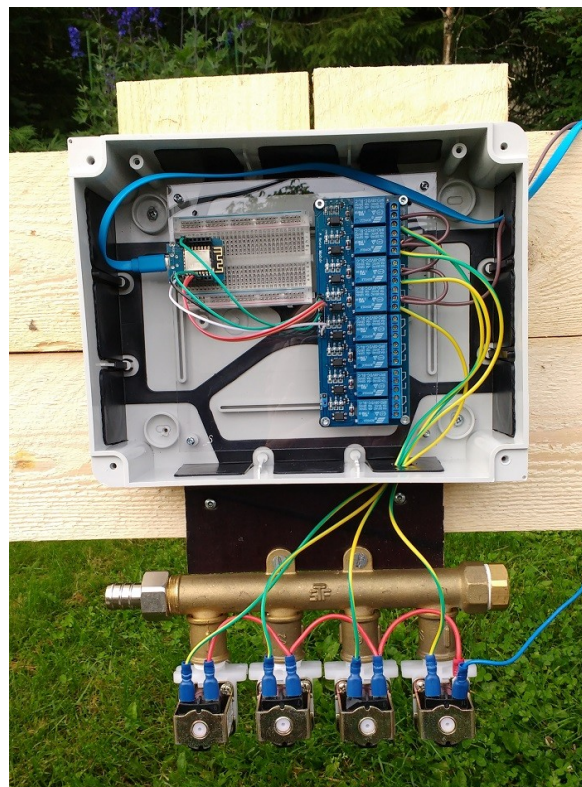
Virrakytkentä- ja kastelunohjausyksikkö

Virrankytkentäyksikkö ja kastelunohjausyksikkö toimivat samalla tavalla. Nissä on nodeMCU-yksikkö, jolla ohjataan releitä halutun toiminnon kytkemiseksi. Virrankytkentäyksikkönä käytetään katkaisimella ohjattavaa pistorasiaa, jossa oleva mekaaninen katkaisin ohitetaan lisäämällä johdotus releohjattun katkaisijaan.



Kuva 4: Virtakatkaisin

Kasteluyksikössä on neljä hanaa, joita voidaan ohjata sulkea ja käynnistää halutulla tavalla.



Kuva 5: Kastelunohjausreleet ja ohjausyksikkö

Kastelunohjausyksikön virtalähteet ovat toisella puolella. Releet ovat 12 voltin releitä, ja saavat virran verkkovirrasta 12 voltin muuntajalla. NodeMCU-yksikkö saa virtansa tavallisesta matkapuhelimen laturista.

Kastelujärjestelmän ohjausohjelma

Jatkokehitystarpeita

Tavoittena on rakentaa Raspberry Pi 3 -pohjainen täysin automatisoitu järjestelmä, joka säätelee kastelua tarpeen mukaan ja huolehtii kasvihuoneen ilmanvaihdesta säätämällä tuuletusluukkujen aukipitoa lämpötilan mukaan. Sään tarkkailua varten lisätään myöhemmin valon määrää, ilmanpainetta, tuulta ja salamoita havainnoivat anturit.