**Edupoli**

**TIETO-JA TIETOLIIKENNETEKNIIKAN PERUSTUTKINTO**

**Sulautetut sovellukset ja projektityöt,30 osp**

Näyttöprojektin kuvaus

Crowtail-automaattisen kastelujärjestelmän kasaus ja skemojen piirto.

**Mikko Harjama**

**17.12.2018**

1 Ammattitaitovaatimukset 1

2 Kuvaus laitteesta 1

3 Kuvaus elektroniikasta 2

4 Kuvaus ohjelmasta 3

5 Linkit, ohjeet, inspiraatiolähteet 4

6 (YTO Kestävän kehityksen edistäminen) 5

7 Projektikansio 6

* Kuvaus elektroniikasta

<https://www.partco.fi/fi/rakennussarjat/crowtail/19684-ard-aak90039k.html?search_query=kastelu&results=2>

Crowtail-kastelujärjestelmäpaketti sisältää yhden Crowtail Smart Pum Board V2.0 -kortin, neljä Crowtail mullan kosteusanturia, yhden nelikanavaisen venttiilin, yhden Crowtail vesipumpun, neljä 50cm Crowtail johtoa, 5,5m vesiletkun ja yhden 12VDC 2A seinäadapterin. Crowtail-kastelujärjestelmän toimintaperiaate on mitata maaperän kosteutta ja antaa vettä oikea määrä, kun tietty maaperän kuivuustaso on saavutettu. Alla kytkentäkaavio.



* Kuvaus ohjelmasta

Koodiin sisältyy 16x2 näytön kirjasto.

#include <LiquidCrystal.h>

// Kosteusanturien (analogi)pinnien määrittäminen

int moisture1 = A0;

int moisture2 = A1;

int moisture3 = A2;

int moisture4 = A3;

// Kosteusanturin arvo

int moisture1\_value = 0;

int moisture2\_value = 0;

int moisture3\_value = 0;

int moisture4\_value = 0;

//Kosteusanturien arvojen muuntaminen prosenteiksi

int percentValue1 = (moisture1\_value)\*100L/(1023);

int percentValue2 = (moisture2\_value)\*100L/(1023);

int percentValue3 = (moisture3\_value)\*100L/(1023);

int percentValue4 = (moisture4\_value)\*100L/(1023);

// Releiden pinnien määrittäminen

int relay1 = 3;

int relay2 = 4;

int relay3 = 5;

int relay4 = 6;

// Vesipumpun pinnin määrittäminen

int pump = 2;

//Näytön ja näytön taustavalon pinnien määrittäminen

const int rs = 7, e = 8, d4 = 9, d5 = 10, d6 = 11, d7 = 12;

LiquidCrystal lcd(rs, e, d4, d5, d6, d7);

const int lcdBacklight = 13;

void setup() {

// Ulostulojen määrittäminen

pinMode(relay1, OUTPUT);

pinMode(relay2, OUTPUT);

pinMode(relay3, OUTPUT);

pinMode(relay4, OUTPUT);

pinMode(pump, OUTPUT);

// Sarjaportin määrittäminen

Serial.begin(9600);

//Taustavalon ulostulon määrittäminen

pinMode(lcdBacklight, OUTPUT);

digitalWrite(lcdBacklight, HIGH);

//Tervehdystekstin määrittäminen

lcd.begin(16, 2);

lcd.print("Hello beautiful!");

delay(2000);

lcd.clear();

//Käynnistys

//3

lcd.setCursor(0, 0);

lcd.print("Hello beautiful!");

lcd.setCursor(7, 1);

lcd.print("-3-");

delay(1000);

lcd.clear();

//2

lcd.setCursor(0, 0);

lcd.print("Hello beautiful!");

lcd.setCursor(7, 1);

lcd.print("-2-");

delay(1000);

lcd.clear();

//1

lcd.setCursor(0, 0);

lcd.print("Hello beautiful!");

lcd.setCursor(7, 1);

lcd.print("-1-");

delay(1000);

lcd.clear();

}

void Kasvit(){

lcd.clear();

//Kasvi 1

moisture1\_value = analogRead(moisture1);

Serial.print("Anturi1: ");

Serial.print(moisture1\_value);

Serial.print(" / ");

Serial.print(percentValue1);

Serial.println("%");

lcd.setCursor(0, 0);

lcd.print("K1");

lcd.setCursor(4, 0);

lcd.print(percentValue1);

lcd.print("%");

//Kasvi 2

moisture2\_value = analogRead(moisture2);

Serial.print("Anturi2: ");

Serial.print(moisture2\_value);

Serial.print(" / ");

Serial.print(percentValue2);

Serial.println("%");

lcd.setCursor(8, 0);

lcd.print("K2");

lcd.setCursor(12, 0);

lcd.print(percentValue2);

lcd.print("%");

//Kasvi 3

moisture3\_value = analogRead(moisture3);

Serial.print("Anturi3: ");

Serial.print(moisture3\_value);

Serial.print(" / ");

Serial.print(percentValue3);

Serial.println("%");

lcd.setCursor(0, 1);

lcd.print("K3");

lcd.setCursor(4, 1);

lcd.print(percentValue3);

lcd.print("%");

//Kasvi 4

Serial.print("Anturi4: ");

Serial.print(moisture4\_value);

Serial.print(" / ");

Serial.print(percentValue4);

Serial.println("%");

Serial.println(" ");

lcd.setCursor(8, 1);

lcd.print("K4");

lcd.setCursor(12, 1);

lcd.print(percentValue4);

lcd.print("%");

Anturi();

}

//Kosteusanturit

void Anturi(){

if(moisture1\_value<=450){

digitalWrite(relay1, HIGH);

lcd.setCursor(2, 0);

lcd.print("!");

}

if(moisture2\_value<=450){

digitalWrite(relay2, HIGH);

lcd.setCursor(10, 0);

lcd.print("!");

}

if(moisture3\_value<=450){

digitalWrite(relay3, HIGH);

lcd.setCursor(2, 1);

lcd.print("!");

}

if(moisture4\_value<=450){

digitalWrite(relay4, HIGH);

lcd.setCursor(10, 1);

lcd.print("!");

}

}

// Varmistetaan, että ainakin yksi kasvi tarvitsee vettä

// jos tarvitsee, käynnistetään moottori

void Kastelu(){

if(moisture1\_value<=450 || moisture2\_value<=450 || moisture3\_value<=450 || moisture4\_value<=450){

digitalWrite(pump, HIGH);

}

// Annetaan kasville 5 sekuntia vettä

delay(5000);

// Sammutetaan pumppu

digitalWrite(pump, LOW);

// Suljetaan jokainen venttiili

digitalWrite(relay1, LOW);

digitalWrite(relay2, LOW);

digitalWrite(relay3, LOW);

digitalWrite(relay4, LOW);

}

void loop() {

Kasvit();

Kastelu();

// Odotetaan () sekuntia ja toistetaan prosessi

delay(1000);

}

* Linkit, ohjeet, inspiraatiolähteet

Kastelujärjestelmän parissa työnskentely oli opettavasita ja antoisaa. Koodaus puolella lähes kaikki onnistui suhteellisen hyvin luokkatovereiden auttaessa tiukan paikan tullen. Varsinkin vierustoveri Toni oli suuri apu tässä projeksissa. Haastavinta oli kytkentäkaavojen piirtäminen oikeiden osien puuttuessa. Esimerkiksi Crowtail smart pum-kortille ei löytynyt kirjastoa Eagleen, joten piti hieman soveltaa, että kytkentäkaavion sai tehtyä.

* (YTO Kestävän kehityksen edistäminen)

[https://eperusteet.opintopolku.fi/#/fi/esitys/3397335/reformi/tutkinnonosat/4209965](https://eperusteet.opintopolku.fi/)

Pakolliset osaamistavoitteet, 1 osp

Opiskelija osaa

* toimia kestävän kehityksen periaatteiden mukaisesti
* ottaa huomioon elinkaariajattelun periaatteet
* pohtia ratkaisuja eettisistä näkökulmista.

Mieti elektronisen laitteen elinkaarta. Mistä raaka-aineet? Eri komponenttien valmistusprosessia. Laitteen valmistusprosessia (piirilevy, kotelo, pakkaus, teholähteet). Hajonneen laitteen kierrätys.

Tämän laitteen heikko lenkki on kosteusantureissa sijaitseva kupari, jonka kuluminen saattaa lyhentää anturien elinkaarta. Laitetta ei kannata pitää jatkuvasti mittaustilassa vaan mitata kosteutta kerran tai kaksi, korkeintaan muutamia kertoja päivässä. Usein pelkästään aamulla tai aamulla ja päivällä/illalla on riittävä määrä kosteuden mittaamiselle, jos maaperän kosteuden jatkuvalle tarkkailulle ei ole tarvetta.

Laitteen alkuperämaata en saanut selville, mutta voisin olettaa, että komponentit ja itse laite valmistetaan jossain Kiinan kupeessa lähes kokonaaan. Tältä osin en osaa sanoa alkuperämaan kierrätystavoista tai heidän työskentelymenetelmistään.

Laite on suurelta osin tehty muovista ja erilaisista elektronisista komponenteista. Täällä Suomessa voimme kierrättää komonentteja kierrätyskeskuksissa sijaitsevilla elektroniikan keräyspisteillä. Puhtaat muovit ja kumitkin voi kierrättää helposti tänä päivänä. Esimerksi varsin monissa talonyhtiöissä on jo suhteellisen kattavat tilat kierrättämiselle.

* Projektikansio

<https://github.com/slerbalenkki/Automatic_plant_watering>

* komponenttiluettelo (BOM)
* kytkentäkaavio
* piirilevysuunnitelma
* ohjelmiston lähdekoodit
* tämä dokumentti (Word)