TEKNILLINEN TIEDEKUNTA

LAITTEEN TILASEURANTA PILVIPALVELUUN

Eetu Karvonen

KONETEKNIIKAN TUTKINTO-OHJELMA

Kandidaatintyö

Kesäkuu 2020

TIIVISTELMÄ

Laitteen tilaseuranta pilvipalveluun

Eetu Karvonen

Oulun yliopisto, Konetekniikan tutkinto-ohjelma

Kandidaatintyö 2020, 26 s. + 2 liitettä

Työn ohjaaja: Yrjö Louhisalmi

Laitteiden tilaa halutaan monesti seurata etänä internetin välityksellä. Tässä työssä suunnitellaan järjestelmä, jonka avulla voidaan seurata laitteiden käyttöä pilvipalvelun avulla sekä lähettää laitteiden käynnistymisestä ilmoituksia. Järjestelmässä käytetään Arduino -pohjaista NodeMCU -kehityskorttia sekä Ubidots -pilvipalvelua.

Avainsanat: IoT, Arduino, pilvipalvelu, Ubidots, laiteseuranta

ABSTRACT

Monitoring of the machine in the cloud

Eetu Karvonen

University of Oulu, Degree Programme of Mechanical Engineering

Bachelor's thesis 2020, 26 p. + 2 appendixes

Supervisor: Yrjö Louhisalmi

State of the machines can be monitored remotely via the internet. In this study will be designed a system that can be used to monitor machines and also send a notification when machine is turned on. Arduino based NodeMCU development board and Ubidots cloud service are used in the system.

Keywords: IoT, Arduino, cloud, Ubidots, monitoring of the machinces

ALKUSANAT

Tämä työ on tehty osana tekniikan kandidaatin tutkintoa Oulun yliopistossa. Työ on valittu käytännönläheisen ja mielenkiintoisen aiheen vuoksi.

Kiitokset Yrjö Louhisalmelle työn ohjauksesta. Hän auttoi mielenkiintoisen aiheen löytämisessä ja opasti työn teossa.

Oulu, 5.6.2020

Eetu Karvonen

Sisällysluettelo

LYHENTEET JA MERKINNÄT

GPIO	General Purpose Input/Output, yleiskäyttöinen sisään-/ulostulo
I	Sähkövirta
R	Resistanssi
U	Jännite

1 JOHDANTO

Oulun yliopiston pajalla on käytössä jyrsinkone, jonka käyttöä halutaan valvoa. Mikäli joku sattuisi käynnistämään laitteen ilman lupaa, tapahtumasta tulee sähköposti-ilmoitus. Tällä hetkellä yliopistolla on käytössä kaupallisella ohjelmoitavalla logiikalla toteutettu järjestelmä. Tämän työn tarkoituksena on korvata nykyinen järjestelmä yksinkertaisemmalla, halvemmalla sekä fyysisesti pienemmällä ratkaisulla.

Työn tarkoituksena on tehdä järjestelmä, jolla saadaan esimerkiksi jyrsinkoneen tila (on/off) pilvipalveluun. Työ sisältää järjestelmän suunnitelun ja käytännön toteutuksen.

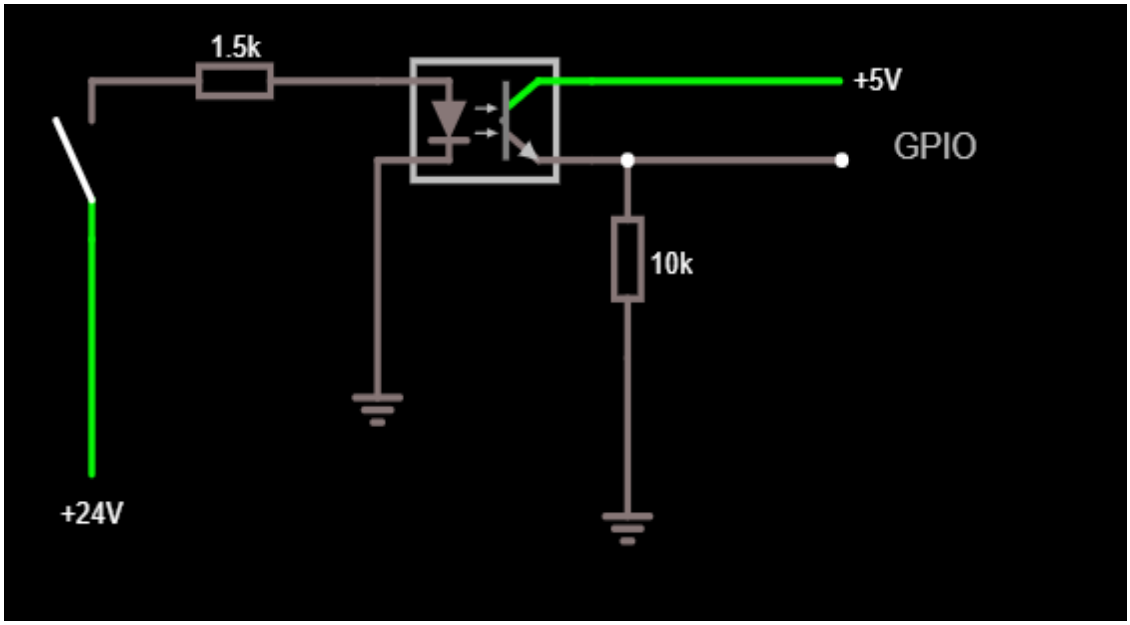
2 JÄRJESTELMÄN SUUNNITTELU

Työssä halutaan saada jyrsinkoneen tilatieto seurantaan (päällä/pois päältä) internetin välityksellä. Tilatieto jyrsinkoneelta tulee 24 voltin jännitesignaalina. Tämä jännitesignaali tulee muuntaa valitulle järjestelmälle sopivaksi. Kun jännitesignaali on muunnettu sopivaksi, signaali ohjataan valitun tietokoneen yleiskäyttöisen sisään-/ulostulon nastaan (=GPIO). Näin tilatieto saadaan kaapattua esimerkiksi Raspberry Pi -korttitietokoneelle, joka voidaan ohjelmoida päivittämään tieto pilvipalveluun.

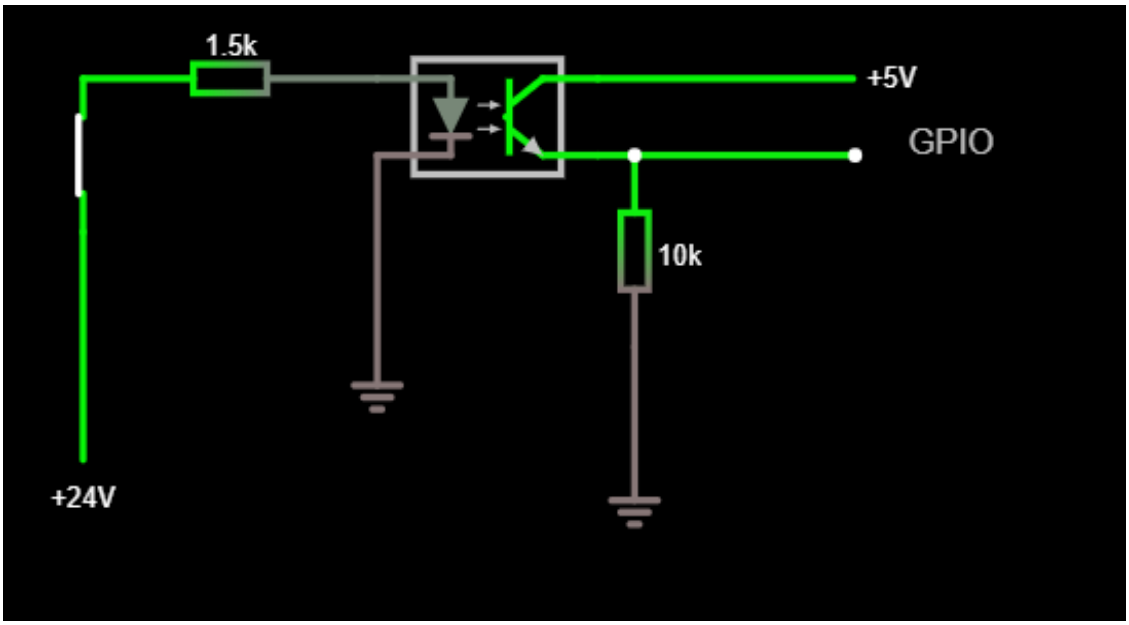
Jännitesignaalin muuttaminen sopivaksi toteutetaan käyttäen optoerotinta. Optoerotin on piiri, jonka sisääntulossa on ledi ja ulostulossa on fototransistori. Kun ledi kytketään palamaan, fototransistori havaitsee ledin aiheuttaman valon ja alkaa johtaa. Näin ollen fototransistoria voidaan ohjata kytkemällä lediä päälle ja pois. Etuna optoerottimen käytöstä on, että 24 voltin jännitesignaali voidaan eristää täysin galvaanisesti matalammasta jännitetasosta. Myös mahdollinen 0 -potentiaaliero ei aiheuta vaaratilanteita. (Kohtala 2007: 16)

2.1 Kytkennän toimintaperiaate

Optoerottimen sisääntuloa voidaan ohjata jyrsinkoneelta tulevalla 24 voltin jännitesignaalilla. Fototransistori johtaa, kun jyrsinkone on päällä. Fototransistorin avulla saadaan jyrsinkoneen tilatieto kytkemällä fototransistorin kollektoriin korttitietokoneen käyttöjännite ja emitteriin korttitietokoneen GPIO nasta, josta digitaalinen tilasignaali luetaan. Kun jyrsinkone on päällä, optoerottimen ledi palaa, fototransisotri johtaa ja korttitietokoneen käyttöjännite nostaa GPIO nastan tilan ylös. Kun jyrsinkone sammutetaan, optoerottimen ledi sammuu, fototransistori ei enää johda eikä korttitietokoneen käyttöjännite pääse enää nostamaan GPIO nastan tilaa ylös. Alasvetovastus varmistaa, että GPIO nastan tila vedetään alas. Järjestelmän kytkentäkaavio on kuvattuna alla olevissa kuvissa (kuvat 1 ja 2).



Kuva 1. Järjestelmän kytkentäkaavio, jyrsinkone ei päällä.



Kuva 2. Järjestelmän kytkentäkaavio, jyrsinkone päällä.

Kytkentäkaaviot on tehty käyttäen piirien simulointityökalua, joka toimii suoraan nettiselaimessa. Työkalu löytyy osoitteesta www.falstad.com/circuit/ .

2.2 Komponenttien valinta

Valitaan optoerottimeksi 4N25 yleiskäyttöinen optoerotin. Optoerottimen datalehti löytyy liitteestä 1. Kyseisen optoerottimen sisääntuloledin käyttöjännite on 1.25 V - 1.50 V ja virrankulutus 10 mA - 60 mA (Components101, 2020). Jyrsinkoneelta tuleva 24 voltin jännitesignaali on sovitettava optoerottimen sisääntuloon sopivalla vastuksella.

Vastus voidaan mitoittaa sopivaksi käyttäen ohmin lakia $U = R \cdot I$, jossa U on jännite, R on resistanssi ja I on sähkövirta. Valitaan sähkövirran suuruudeksi I = 15 mA. Jännite vastuksen yli on U = 24 V – 1.5 V = 22.5 V. Sijoittamalla U ja I saadaan vastuksen suuruudeksi R = 22.5 V / 0.015 A = 1500 Ohm.

2.3 Pilvipalvelu

Aikaisemmin Oulun yliopistolla on ollut käytössä ”Ubidots” -niminen pilvipalvelu. Ubidots tarjoaa pilvipalveluita maksua vastaan yritysasiakkaille sekä ”STEM” -nimeä kantavan ilmaisen palvelun esimerkiksi opiskelijoille ja tutkijoille (Ubidots 2020).

Ubidotsin soveltuvuutta tähän projektiin lähdettiin selvittämään tutkimalla Ubidotsin dokumentaatiota. Pyrittiin selvittämään, miten helposti jyrsinkoneen tilatieto saadaan ohjelmoitua mikrokontrollerilta pilvipalveluun. Tässä vaiheessa tiedettiin jo, että todennäköisesti mikrokontrolleri tulee olemaan joko Arduino tai Raspberry Pi -pohjainen mikrokontrolleri.

Ubidotsin Github -sivuilta löytyi kattavat dokumentaatiot Ubidotsin käyttöön lukuisilla eri tekniikoilla. Dokumentaatiot löytyivät mm. monille Arduino -korteille sekä Python -ohjelmointikielelle. Todettiin tässä vaiheessa, että Ubidots soveltuu käytettäväksi tähän projektiin. (Github, Ubidots 2019)

2.4 Käytettävän mikrokontrollerin valinta

Vaatimuksena käytettävälle mikrokontrollerille on internetyhteys sekä GPIO-liitäntä. Lisäksi valintaan vaikuttaa käytettävän pilvipalvelun helppo yhteensovitettavuus.

Vaihtoehtoja on käytännössä kaksi: Arduino tai Raspberry Pi -pohjainen.

Halvin Raspberry Pi -kortti on Raspberry Pi Zero W, joka täyttää vaatimukset, mutta on jopa hieman turhankin tehokas. Raspberry Pi pystyy ajamaan omaa käyttöjärjestelmää ja toiminnallisuus olisikin helppo ohjelmoida suoraan itse laitteella esimerkiksi Python -ohjelmointikielellä.

Arduino -pohjaisista mikrokontrollereista sopivin olisi NodeMCU ESP8266 -kehityskortti, josta löytyy sisäänrakennettu WiFi -piiri. Laskentatehoa kyseisessä kortissa on huomattavan paljon vähemmän, kuin Raspberry Pi:ssa, mutta kuitenkin aivan riittävästi. Omaa käyttöjärjestelmää laitteessa ei ole, vaan ohjelmakoodi pitää ladata laitteeseen USB-väylää pitkin esimerkiksi käyttäen Arduino IDE -kehitysympäristöä Windows-, Linux- tai Mac -tietokonella. Ohjelmointi tapahtuu Arduino IDE -kehitysympäristössä C ja C++ -ohjelmointikieliä muistuttavalla Arduinon omalla ohjelmointikielellä. (Margolis 2011)

Raspberry Pi Zero W:n hinta on noin 25 euroa ja NodeMCU ESP8266 korttia saa noin 14 euron hintaan, Kiinasta jopa noin viidellä eurolla. Raspberry Pi:n valintaa puoltaisi laitteen helpompi ohjelmointi, mutta toisaalta vaadittu laskentateho on niin pieni, että Raspberry Pi olisi huomattava ylilyönti. Raspberry Pi:lle pitäisi suunnitella myös paristovarmennus (UPS) ja sammutustoiminto mahdollisten jännitekatkojen varalle. Lisäksi Oulun yliopistolta löytyi runsaasti NodeMCU -kortteja, joten valittiin projektissa käytettäväksi NodeMCU -kortti.

3. LAITTEEN KÄYTTÖÖNOTTO JA TESTAUS

Kun kompnentit on valittu ja järjestelmän toimintaperiaate suunniteltu, voidaan aloittaa testaamaan systeemin toimivuutta käytännössä.

Windows-, Mac- sekä Linux -tietokoneille on saatavilla Arduino IDE -kehitysympäristö, jolla NodeMCU -korttia pystyy ohjelmoimaan (Arduino 2020).

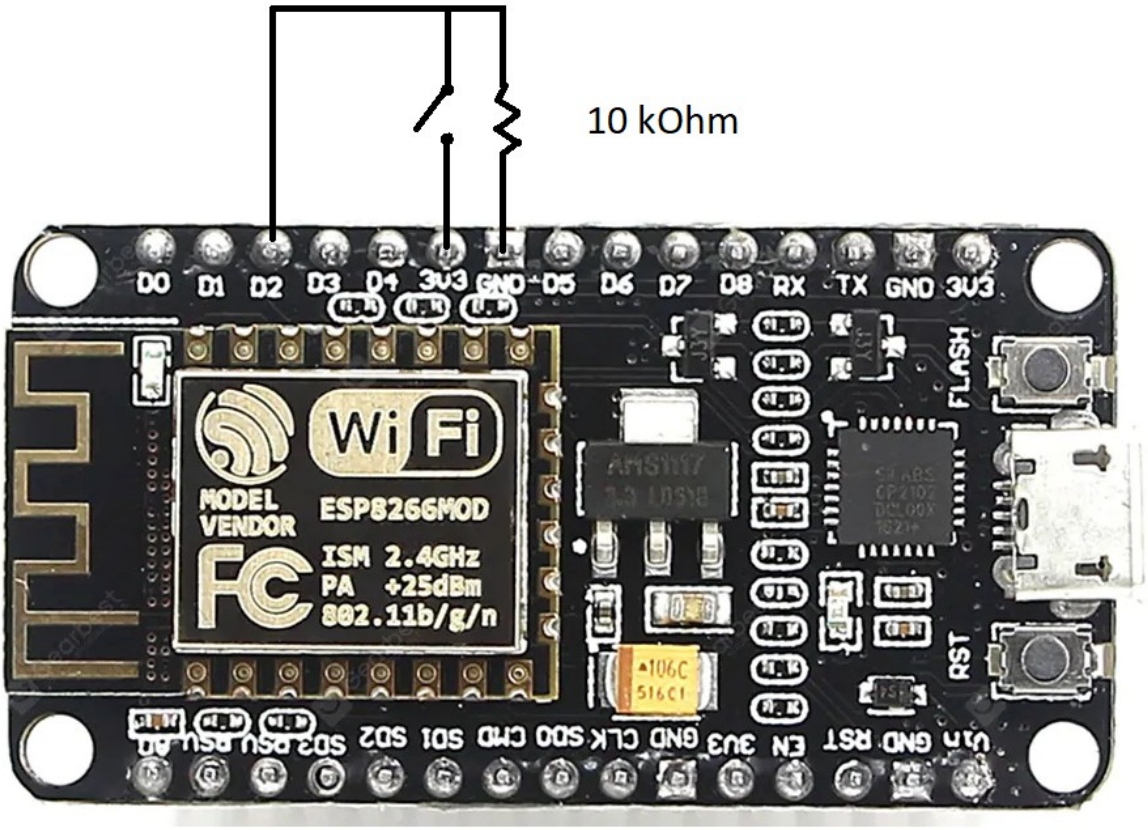
Kehitysympäristö ei suoraan tue NodeMCU -korttia, vaan kehitysympäristöön täytyy ladata tuki NodeMCU -mallille. Asetuksissa kohtaan ”Additional board manager URLs” täytyy ensin lisätä osoite:

”https://arduino.esp8266.com/stable/package_esp8266com_index.json”,

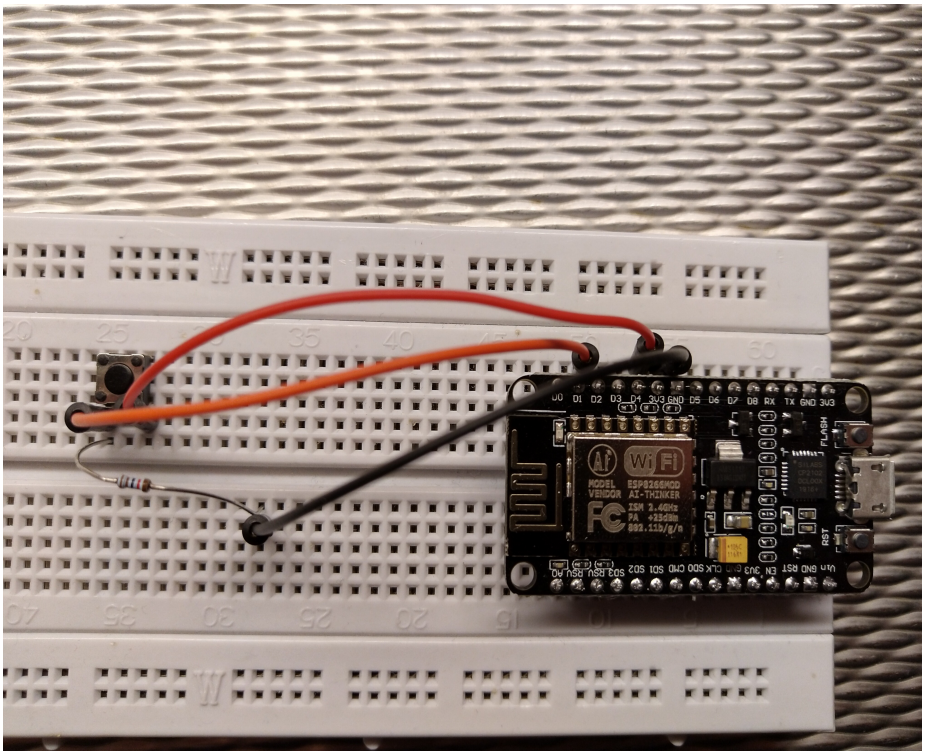
josta kehitysympäristö löytää tarvittavat lisäosat. Tämän jälkeen voidaan ”Board manager” -kohdasta etsiä ”esp8266” -kirjasto ja asentamalla kyseinen kirjasto saadaan NodeMCU -kortille tuki kehitysympäristöön. NodeMCU on vielä valittava käytettäväksi kortiksi. (Github, Ubidots 2019)

Ensimmäisenä testattiin digitaalisen sisääntulon toimintaa painonapin avulla. Testikytkentä tehtiin Arduinon virallisilta nettisivuilta löytyvien ohjeiden mukaan, muokattuna NodeMCU:lle (Arduino 2020). Kuvien 1 ja 2 alustavassa kytkentäkaaviossa GPIO:n jännite on 5 volttia, mutta NodeMCU:n GPIO:n jännite on 3.3 volttia. Myös Arduinon nettisivuilta löytyvän painonappiesimerkin Arduino -laitteen GPIO:n jännite on 5 volttia.

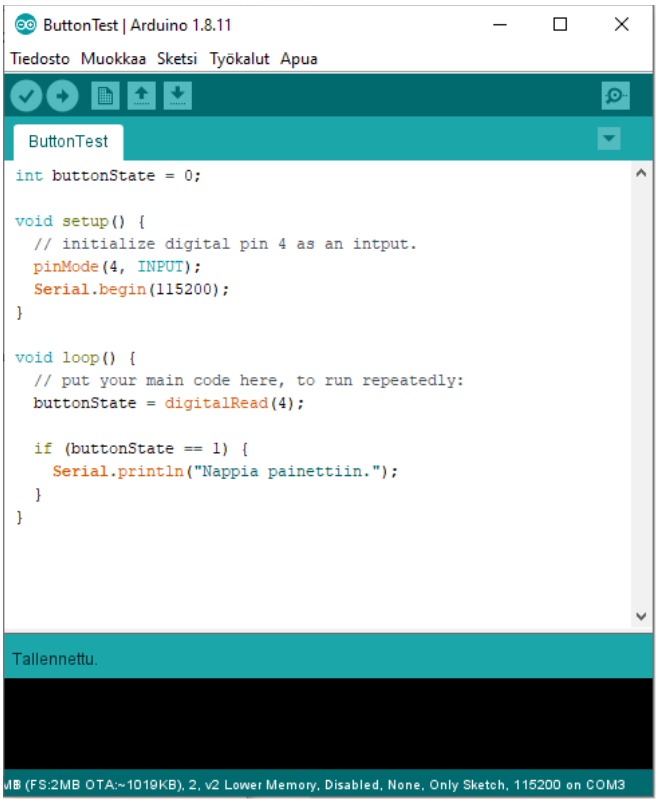
Kytkennässä oleva vastus on niin sanottu alasvetovastus, jonka tehtävänä on varmistaa, että kytkennän looginen tila menee nollaan, kun nappia ei enää paineta. Näin ollen vastuksen mitoitus ei ole tarkkaa ja kytkennässä voidaan käyttää 5 voltin GPIO jännitteen sijaan 3.3 volttia. Vastuksen koko on Arduinon esimerkin mukaisesti 10 kOhm. Arduinon nettisivuilla oleva ohjelma on nimeltään ”DigitalReadSerial.ino”, jota muokattiin kuvan 5 mukaiseksi. NodeMCU:n D2 niminen pinni on Arduinon GPIO numero 4, joka on otettava huomioon ohjelmakoodissa (Esp8266-shop 2020). Kytkentäkaavio on esitetty kuvassa 3 ja kuvassa 4 on valokuva testikytkennästä. Käyttöjännitteen NodeMCU saa USB-liitännästä.



Kuva 3. Kytkentäkaavio painonapin testaamiselle.



Kuva 4. Testikytkentä painonapin testaamiseksi.



Kuva 5. Testiohjelma painonapin testaamiseksi.

Arduino-ohjelman peruskomponentit ovat ”void setup()” ja ”void loop()” -funktiot. Setup -funktioon tulee käskyt, jotka ohjelma suorittaa vain kerran, kun arduino käynnistyy. Loop -funktion käskyt arduino suorittaa jatkuvasti uudestaan niin kauan, kun arduinossa on virrat päällä.

Kuvan 5 testiohjelman setup -funktiossa asetetaan pinni neljä sisääntuloksi ja alustetaan sarjamonitori toimimaan nopeudella 115200. Loop -funktiossa luetaan pinnin neljä tila ja asetetaan tila muuttujaan ”buttonState”. Jos pinnin neljä tila on yksi, eli nappia painetaan, tulostetaan sarjamonitoriin teksti ”Nappia painettiin”.

Testiohjelma ladattiin NodeMCU:lle USB-väylää pitkin ja huomattiin, että nappia painettaessa sarjamonitoriin tulostuu teksti: ”Nappia painettiin.”. Testi siis toimi.

4. PILVIPALVELUN KÄYTTÖÖNOTTO JA TESTAUS

Seuraavaksi testattiin NodeMCU:n ja pilvipalvelun välistä kommunikaatiota. Ubidotsiin rekisteröidytään Ubidotsin nettisivuilla ”www.ubidots.com” antamalla sähköpostiosoite, käyttäjänimi sekä salasana. Kun käyttäjätunnus on rekisteröity, Ubidotsin käyttöliittymä vaikutti ensikertalaiselle hyvin käyttäjäystävälliseltä. Kohdasta ”API Credentials” löytyy käyttäjän ”token”, joka on pitkä merkkijono. Tokeni on käyttäjän yksilöivä tunniste, jonka avulla tarvittavat tiedot saa lähetettyä oikealle Ubidots -käyttäjälle.

Pilvipalvelussa ei tässä vaiheessa tarvitse tehdä muuta kuin ottaa tokeni talteen. Tämä tokeni kirjoitetaan sellaisenaan Arduino-ohjelmaan. Kun laite lähettää tokenin osoitteeseen dataa, Ubidots luo uuden laitteen ja halutut muuttujat automaattisesti ja ne ilmestyvät käyttäjän käyttöliittymään.

Muokkaamalla aikaisemmin esillä ollutta painonapin testausohjelmaa, voidaan testata NodeMCU:n ja Ubidotsin kommunikaatiota. Muokataan ohjelmaa siten, että kun painonappia painetaan laite lähettää Ubidotsille muuttujan ”Napin tila” ja arvon 1.

Ubidotsin Github -sivuilta löytyy selkeät ohjeet esimerkkikoodeineen miten NodeMCU ohjelmoidaan lähettämään dataa. Ensin täytyy ladata Ubidots -kirjasto ja lisätä se Arduinon kehitysympäristöön. Tämän jälkeen voidaan ohjelmoida halutut toiminnallisuudet. Kuvan 6 ohjelmakoodi on tehty käyttäen apuna Ubidotsin esimerkkejä. (Github, Ubidots 2019)

Kuva 6. Testiohjelma Ubidots -kommunikoinnin testaamiseen.

Kuvan 6 testiohjelmasta löytyy tutut setup ja loop -funktiot. Ennen setup -funktiota ohjelman ensimmäisellä rivillä ohjelmaan otetaan mukaan Ubidots -kirjasto. Tämän jälkeen seuraavilla riveillä ennen setup -funktiota luodaan muuttujat, joita ohjelmassa tarvitaan. Muuttujaan ”ssid” täytyy laittaa WiFi -verkon nimi, johon halutaan yhdistää ja ”pw” muuttujaan WiFi -verkon salasana. Käytetty WiFi -verkko oli ”panoulu”, joka ei vaadi salasanaa. Mikäli salasanaa ei ole, muuttujaan ”pw” laitetaan tyhjä merkkijono (pelkät lainausmerkit). ”UBIDOTS_TOKEN” -muuttujaan tulee Ubidots -tokeni, jonka saa kirjautumalla Ubidotsin käyttöliittymään.

Setup -funktiossa asetetaan taas pinni neljä sisääntuloksi ja alustetaan sarjamonitori toimimaan nopeudella 115200. Lisäksi käytetään Ubidots -objektia yhdistämään langattomaan verkkoon ja asetetaan debuggaus moodi päälle. Delay käsky käskee ohjelmaa odottamaan paikallaan 10 millisekuntia.

Loop -funktiossa luetaan taas pinnin neljä tila ja asetetaan tila muuttujaan ”buttonState”. Jos pinnin neljä tila on yksi eli nappia painettiin, ohjelma lähettää Ubidotsiin muuttujan ”Napin tila”, jonka arvo on yksi. Käsky ”ubidots.add()” lisää muuttujan lähetyspuskuriin ja ”ubidots.send()” -käsky lähettää puskurin. Mikäli ”ubidots.send()” palauttaa ”true”, eli lähetys onnistui, sarjamonitori tulostaa tesktin ”Values sent”.

Ohjelma ladattiin NodeMCU:lle ja huomattiin, että napin painamisen jälkeen Ubidotsin käyttöliittymään ilmestyi uusi laite ja muuttuja ”Napin tila”, jonka arvo on 1. Testi siis toimi.

Lisäksi testattiin, miten Ubidotsin saa lähettämään sähköposti-ilmoituksia. Ubidots oli jälleen hyvin aloittelijaystävällinen. Sähköposti-ilmoituksia varten löytyi käyttöliittymästä kohta ”events”, josta pystyi luomaan erilaisia tapahtumia. Tapahtumat ovat muotoa ”jos jotain tapahtuu, tee jotain.” Luotiin tapahtuma ”jos 'Napin tila' on 1, lähetä sähköposti-ilmoitus.” Tämän jälkeen kun Ubidots muuttujan ”Napin tila” muuttui ykköseksi, käyttäjän sähköpostiin tuli ilmoitus.

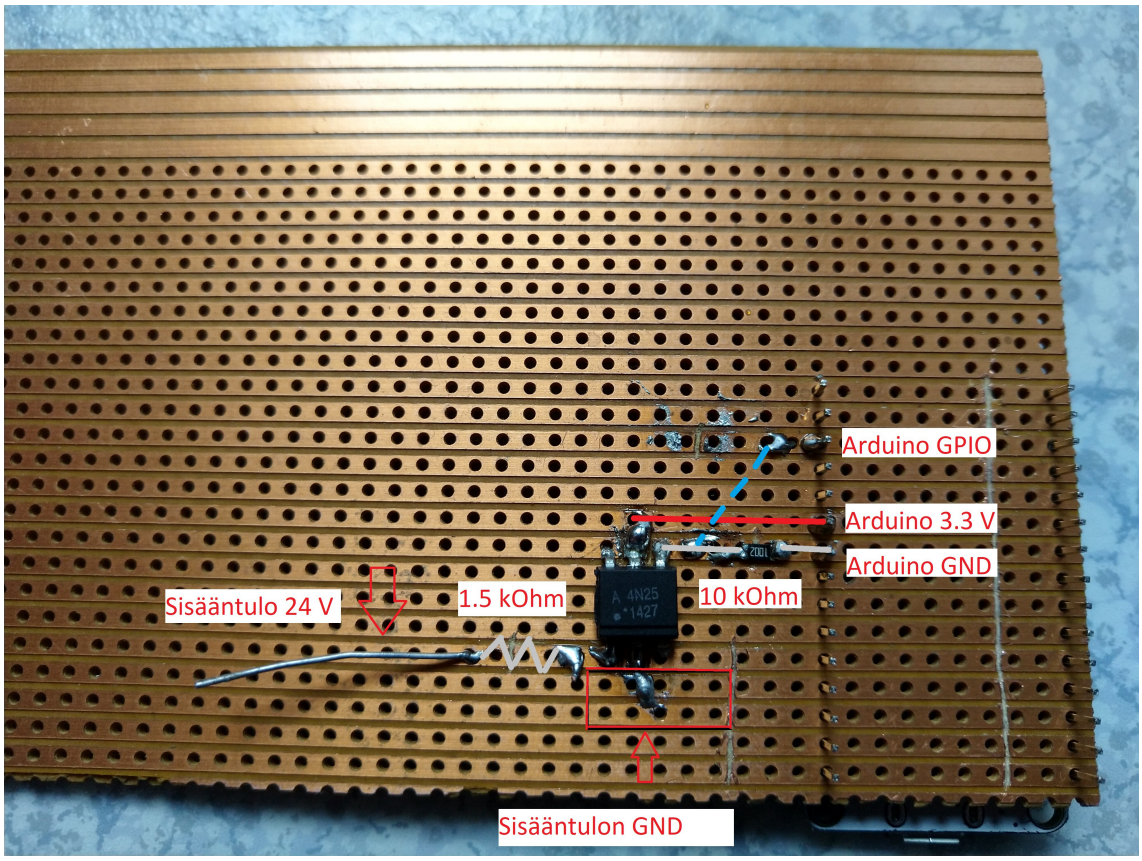
5. SYSTEEMIN KOKOONPANO JA TESTAUS

Seuraavaksi pohdittiin, miten järjestelmän saisi liitettyä osaksi jyrsinkonetta. Jyrsinkonetta käytiin tarkastelemassa paikanpäällä ja huomattiin, että jyrsinkoneen päälläolosta kertova signaali tulee yksinkertaisesti tavallista sähköjohtoa pitkin.

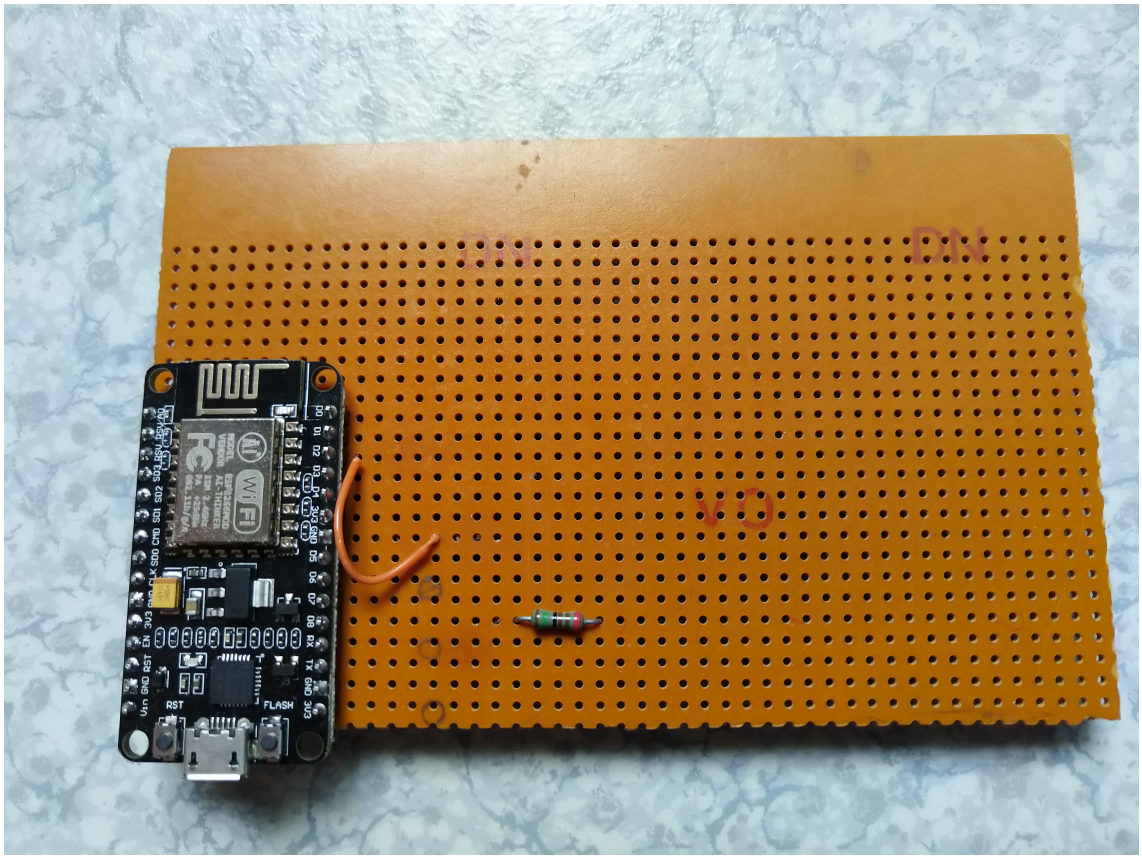
Yksinkertaisin tapa toteuttaa kokoonpano on suunnitella piirilevy ja juottaa komponentit siihen. Piirilevyn liittäminen jyrsinkoneen tilasignaalin johtoihin voidaan toteuttaa esimerkiksi juottamalla johdot suoraan piirilevylle tai juottamalla piirilevylle liittimet, joilla johdot voidaan helposti liittää piirilevyyn.

Piirilevynä päätettiin käyttää reikäkytkentälevyä, johon voi nopeasti juottaa pienen määrän komponentteja ilman, että tarvitsee suunnitella ja valmistaa erillistä piirilevyä. Suositeltavaa kuitenkin olisi käyttää esimerkiksi EAGLE -ohjelmistoa ja valmistaa käyttötarkoitukseen räätälöity piirilevy.

Komponentit juotettiin reikäkytkentälevylle luvussa 2 olleen suunnitelman ja kuvan 1 piirikaavion mukaisesti. Kuvassa 7 komponentit on juotettu kytkentälevylle. Kuvasta näkyy hyvin, ettei kytkentälevylle juottaminen ole optimiratkaisu, vaan käyttötarkoitusta varten valmistettu piirilevy olisi parempi vaihtoehto. Kytkentälevy sattui olemaan mallia, jossa rivit ovat yhteen kytkettyinä. Tämän takia jouduttiin erottamaan tarvittavat rivit toisistaan.



Kuva 7. Komponentit juotettuna reikäkytkentälevylle.



Kuva 8. Reikäkytkentälevyn toinen puoli.

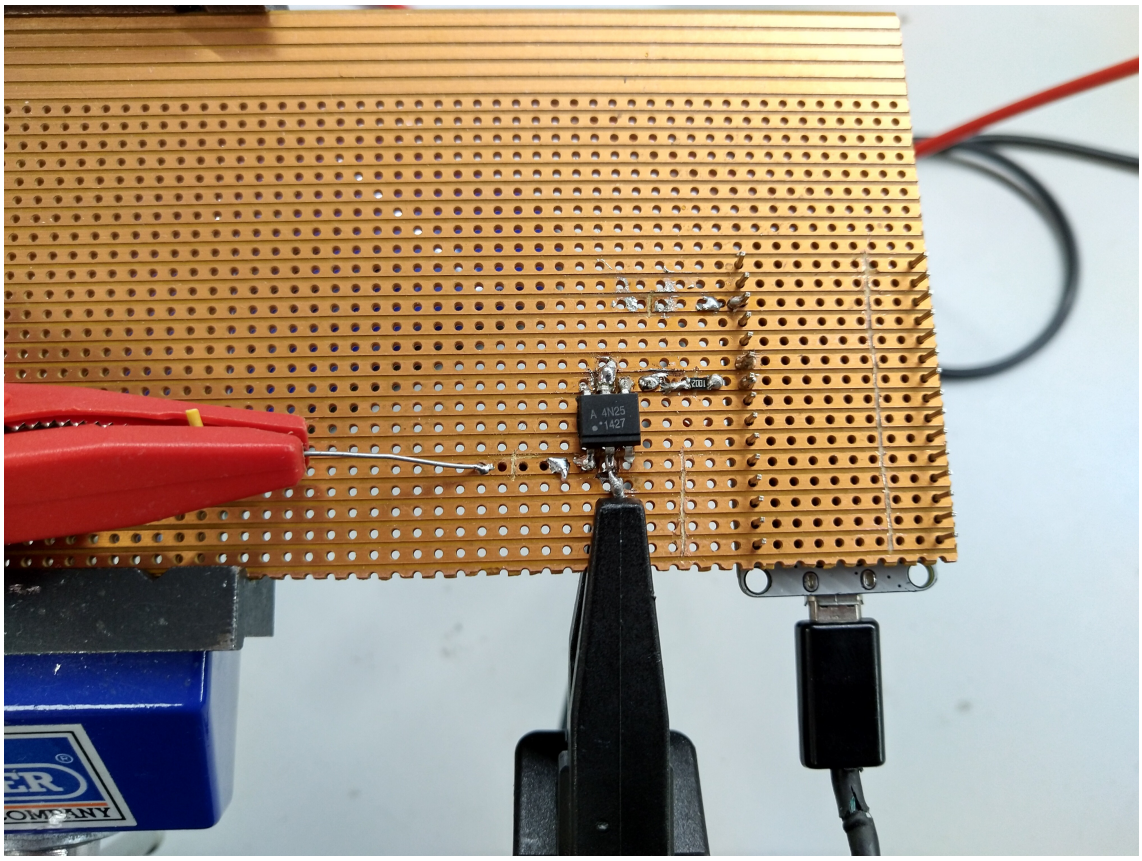
Optoerotin ja 10 kOhmin vastus ovat pintaliitoskomponentteja ja 1,5 kOhmin vastus jalallinen. 1,5 kOhmin vastuksen ulompi jalka jätettiin toistaiseksi pitkäksi testaamista varten. Lisäksi johdon avulla on yhdistetty NodeMCU:n D2 nasta optoerottimen emitteriin (kuva 8).

Piiriin tulee siis 24 voltin jännitesignaalin plusjohto 1,5 kOhmin vastuksen uloimpaan jalkaan ja maataso optoerottimen sisääntulopuolen keskimmäiseen jalkaan. Erilliset liittimet päätettiin jättää juottamatta ajan säästämisen vuoksi.

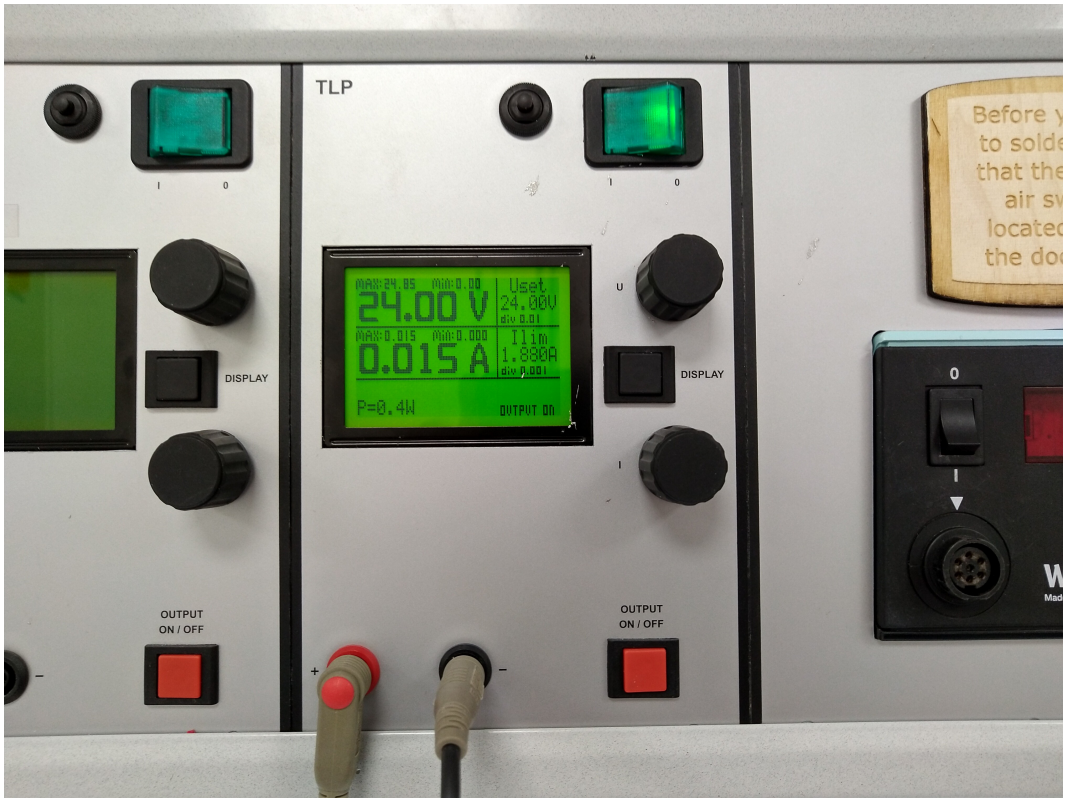
5.1 Optoerottimen testaus

Systeemin toimivuutta testattiin ulkoisen virtalähteen avulla. Virtalähde mallintaa testissä jyrsinkoneelta tulevaa 24 voltin tilasignaalia. NodeMCU tarvitsee toimiakseen 5 voltin käyttöjännitteen, joka voidaan syöttää laitteessa olevaan micro-USB -porttiin. Testattaessa virta NodeMCU:lle otettiin tietokoneen USB-portista, jota käytettiin myös ohjelman lataamiseen NodeMCU:lle.

Ensin kokoonpanoa testattiin kytkemällä pelkästään virtalähteeltä tuleva 24 volttia optoerottimen sisääntuloon, mutta jättämällä NodeMCU ilman käyttöjännitettä. Näin siis testattiin pelkästään optoerottimen sisääntuloledin päällekytkemistä. Virtalähde säädettiin 24 volttiin ja huomattiin, että päälle laitettaessa virran suuruus oli 15 mA, joka on täsmälleen saman suuruinen, kuin mitä käytettiin vastuksen mitoitusvaiheessa. Vaikutti siis siltä, että optoerottimen sisääntuloledi kytkeytyi odotetusti päälle ja testi toimi odotusten mukaisesti. Kuvissa 9 ja 10 on virtalähteen kytkennät sekä virtalähteen näyttö.



Kuva 9. Virtalähteen kytkennät. USB-johto ei ole kytketty.



Kuva 10. Laboratoriovirtalähteen näyttö.

5.2 Järjestelmän testaus

Lopuksi testattiin koko järjestelmän toimintaa. NodeMCU:lle ladattiin kuvan 11 mukainen ohjelma. Ohjelma on hyvin samankaltainen, kuin pilvipalvelun testaamiseen painonapin avulla käytetty ohjelma. Erona on, että uusi ohjelma vertaa onko GPIO pinnin tila eri, kuin Ubidotsissa oleva tieto. Tähän vertailuun on käytetty muuttujaa ”cloudState”. Ohjelma lukee jatkuvasti GPIO nastan tilaa. Jos ”pinState” on eri kuin ”cloudState”, ohjelma lähettää Ubidotsin muuttujaan ”OnOff” GPIO nastan sen hetkisen tilan. Jos lähetys onnistuu, ohjelma vaihtaa ”cloudState” muuttujan tilaa.

Muuttujaa ”cloudState” käytetään siis pitämään muistissa Ubidotsin tilaa. Näin voidaan verrata, onko Ubidotsin tila eri kuin GPIO nastan tila ja voidaan lähettää GPIO nastan tila Ubidotsiin vain tarvittaessa.

Järjestelmän testaus tehtiin kuvan 9 mukaisella kytkennällä. Testikokoonpano on siis muuten samanlainen kuin pelkän optoerottimen testauksessa, mutta tällä kertaa myös NodeMCU:n käyttöjännite on kytketty. Pelkkää optoerotinta testattaessa, testi toimi niin kuin pitikin, mutta kun NodeMCU otettiin testiin mukaan, optoerotin vaikutti menneen rikki.

Kun virtalähteeseen laitettiin virrat päälle, virta lukema värähti hetkellisesti ja meni sitten nollaan. Tämän jälkeen yritettiin useamman kerran ottaa virrat pois päältä, tarkistaa kytkennät ja laittaa virrat päälle. Virtalähde näytti vain nollavirtaa. Vaikutti siltä, kuin optoerottimen sisääntuloledi olisi palanut rikki. Testi ei siis toiminut.

systemTest | Arduino 1.8.11

Tiedosto Muokkaa Sketsi Työkalut Apua

systemTest

```
#include "Ubidots.h"

const char* ssid = "panoulu";                 // Wifi to connect
const char* pw = "";                          // Password for wifi
const char* UBIDOTS_TOKEN = "[redacted]";
Ubidots ubidots(UBIDOTS_TOKEN, UBI_HTTP);

int pinState = 0;
int pin = 4;
int cloudState = 0;

void setup() {
  pinMode(pin, INPUT);
  Serial.begin(115200);
  delay(10);
  ubidots.wifiConnect(ssid, pw);
  ubidots.setDebug(true);
  delay(10);

}

void loop() {
  pinState = digitalRead(pin);
  if (pinState == 1 && cloudState == 0) {
    float value = 1;
    ubidots.add("OnOff", value);

    bool bufferSent = false;
    bufferSent = ubidots.send();

    if (bufferSent) {
      Serial.println("Value sent");
      cloudState = 1;
    }

  } else if (pinState == 0 && cloudState == 1) {
    float value = 0;
    ubidots.add("OnOff", value);

    bool bufferSent = false;
    bufferSent = ubidots.send();

    if (bufferSent) {
      Serial.println("Value sent");
      cloudState = 0;
    }
  }
  delay(1000);
}
```

Tallennettu.

4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3

Kuva 11. Ohjelmakoodi järjestelmän testaamiseksi.

5.3 Epäonnistuneen testin pohdintaa

Optoerotin meni siis rikki järjestelmän testaamisessa, kun mukaan otettiin NodeMCU. Optoerotinta kytkettiin päälle ja pois ennen hajoamista useita kertoja ilman NodeMCU:ta, jolloin optoerotin vaikutti toimivan oikein. NodeMCU ja optoerottimen ulostulopuoli vaikutti olevan vikaantumisen syynä. Hajoamisen jälkeen optoerotin ei toiminut enää edes ilman NodeMCU:ta.

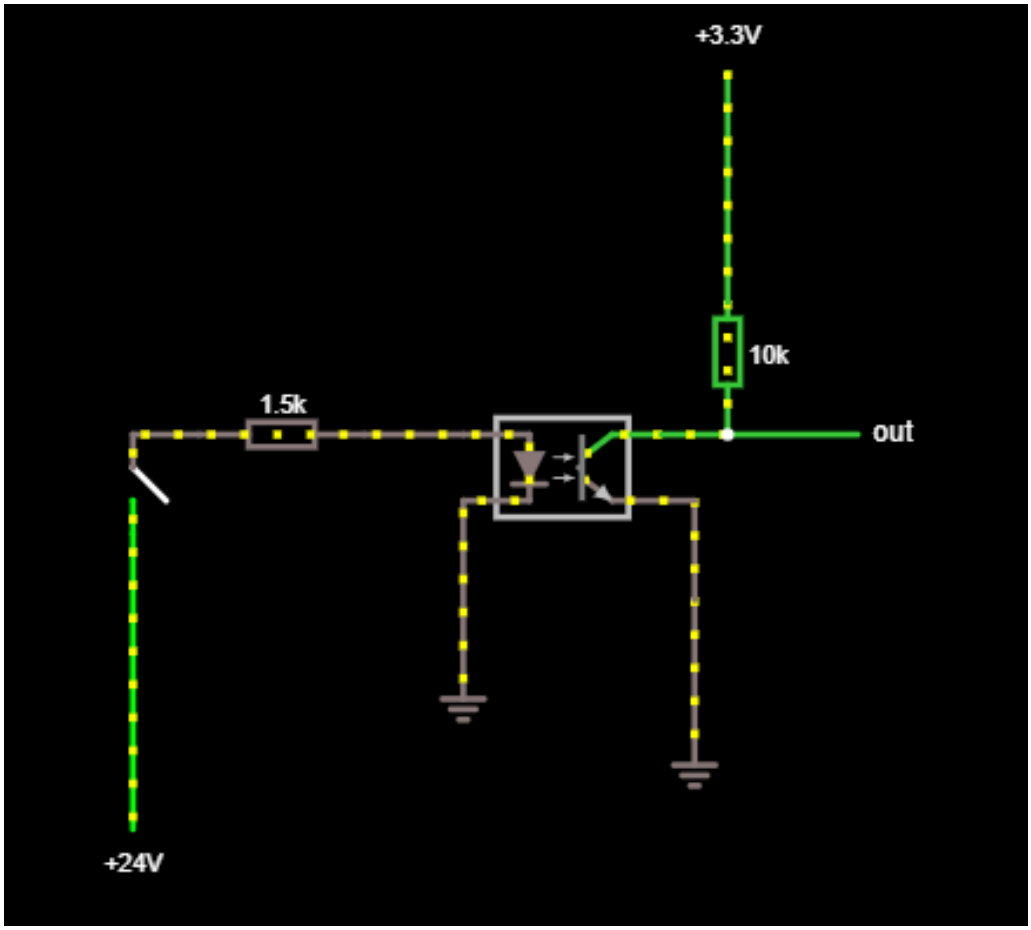
Sisääntulopuoli ja ulostulopuoli on optoerottimessa eristetty, joten herää kysymys, miten ulostulopuolelle kytketty NodeMCU voi aiheuttaa sisääntulopuolen ledin hajoamisen. Näin kuitenkin vaikutti käyvän, koska ainut muutos, mikä testauksessa tehtiin, oli NodeMCU:lle kytketty käyttöjännite.

Optoerottimen ulostulopuolen kytkentä oli sovellettu suoraan arduinon painonappikytkennästä. Optoerottimen ulostulotransistorin toiminta jäljitteli painonappia. Transistorin maksimivirta, joka transistorin läpi sallitaan on paljon pienempi mitä painonapeilla. Käytetyn optoerottimen maksimivirta transistorin läpi on 100 mA (Components 101, 2020). Ohmin lain avulla laskettuna transistorin läpi kulkeva virta kytkennässä on I = U / R = 3.3 V / 10000 Ohm = 0.033 mA eli reilusti alle sallitun maksimivirran.

Vaikka kytkentä jäljittelikin painonapin kytkentää, transistorien kanssa yleisesti käytetty kytkentä on kuvan 12 mukainen. Vastus on siirretty transistorin emitterin ja maan välistä kollektorin ja käyttöjännitteen välille. Lisäksi ulostulo on siirretty transistorin kollektoripuolelle. Tällöin looginen ulostulo muuttuu päinvastaiseksi, joka on otettava huomioon ohjelmakoodissa. Eli sisääntulon ollessa päällä ulostulo on pois päältä ja sisääntulon ollessa pois päältä ulos tulo on päällä.

5.4 Toinen testaus

Rikki mennyt optoerotin vaihdettiin uuteen ja kytkentä muutettiin kuvan 12 mukaiseksi. Lisäksi ohjelmakoodin logiikka muutettiin vastaamaan käänteistä ulostuloa. Uusi ohjelmakoodi löytyy liitteestä 2. Tämän jälkeen testi suoritettiin uudestaan ja huomattiin, että tällä kertaa testi toimi odotusten mukaisesti. Virtalähde näytti 15 mA virtaa ja NodeMCU lähetti muuttujan Ubidotsiin.



Kuva 12. Järjestelmän paranneltu kytkentäkaavio.

6. YHTEENVETO

Tämän työn tarkoituksena oli selvittää, miten esimerkiksi jyrsinkoneen käynnistyessä saadaan tapahtumasta tieto pilvipalveluun sekä sähköposti-ilmoitus. Työssä päädyttiin käyttämään Arduino -pohjaista NodeMCU -kehityskorttia, jonka tehtävänä on lähettää tieto pilvipalveluun internetin välityksellä. Jotta koneen tilatieto saadaan NodeMCU -kortille, täytyy koneen tilasta kertova 24 voltin jännitesignaali muuntaa sopivaksi. Signaalin muuntamiseksi käytettiin optoerotinta. Käytetty pilvipalvelu oli Ubidots, joka osoittautui erittäin helppokäyttöiseksi vaihtoehdoksi.

Järjestelmän ensimmäinen testaus epäonnistui ja rikkoi optoerottimen. Järjestelmän alkuperäistä kytkentäkaaviota muutettiin, jonka jälkeen järjestelmä toimi oikein.

Lopputuloksena saatiin laboratoriovirtalähteen avulla testattu järjestelmä, jota voi käyttää esimerkiksi jyrsinkoneessa. Kustannuksiltaan järjestelmän rakentaminen maksaa alle 20 euroa. Suurin kustannus tulee NodeMCU -kehityskortista, joka maksaa lähteestä riippuen 3-15 euroa. Loput komponentit, vastukset, optoerotin sekä piirilevy, ovat erittäin halpoja.

Työssä käytetty Ubidots -lisenssi on ilmainen, mutta tarjolla on myös kuukausimaksullisia paketteja. Kuukausimaksullisten pakettien hinnat alkavat 49 dollarista kuukaudessa ja kallein paketti on 1799 dollaria kuukaudessa. Kalleimmat paketit on tarkoitettu teolliseen käyttöön ja ne tarjoavat mm. mahdollisuuden suurempaan määrään laitteita ja mahdollistaa suuremman datamäärän keräämisen.

Ubidots -pilvipalvelua testattiin laitteen tilatiedon kirjaamiseen ja sähköposti-ilmoitusten lähettämiseen. Ubidotsia on mahdollista käyttää myös muun datan keräämiseen esimerkiksi kytkemällä NodeMCU:hun antureita. Lisäksi Ubidots -tapahtumat eivät rajoitu pelkästään sähköposti-ilmoituksiin, vaan Ubidotsilla on mahdollista lähettää esimerkiksi tekstiviestejä, Telegram -viestejä sekä HTTP-pyyntöjä.

Järjestelmän testaus tehtiin käyttäen reikäkytkentälevyä, mutta suositeltavaa olisi tehdä laitteelle oma piirilevy luotettavuuden parantamiseksi.

Kokeelliset testaukset tehtiin Oulun yliopiston Fablabissa. Testaamisessa käytettiin avointa panoulu -verkkoa, mutta avoimen verkon käyttö ei ole suositeltavaa tietoturvasyistä.

7. LÄHTEET

Arduino (2020), Recources, Tutorials, Built-in examples, 1. Basics: Digital read serial. https://www.arduino.cc/en/Tutorial/DigitalReadSerial [8.2.2020].

Arduino (2020), Software, Downloads. https://www.arduino.cc/en/Main/Software [7.2.2020].

Components101 (2020), Search, 4n25. https://components101.com/ics/4n25-optocoupler [5.2.2020].

Esp8266-shop (2020), ESP8266 GUIDE, ESP8266 NodeMcu Pinout. https://esp8266-shop.com/esp8266-guide/esp8266-nodemcu-pinout/ [7.2.2020].

Github, Ubidots (2019), Search, Ubidots, esp8266 https://github.com/ubidots/ubidots-esp8266 [7.2.2020].

Kohtala M (2007), Kolmivaiheisen PWM-vaihtosuuntaajan kehittäminen opetuskäyttöön. Tutkintotyö. Tampereen ammattikorkeakoulu. Sähkötekniikan osasto. (https://www.theseus.fi/bitstream/handle/10024/9841/Kohtala.Matti.pdf?sequence=2)

Margolis M (2011), Arduino Cookbook. Sebastopol: O'Reilly

Ubidots (2020), Industries, Stem. https://ubidots.com/stem/ [5.2.2020]

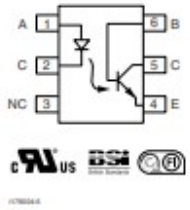
Liite 1. Optoerottimen datalehti.

4N25, 4N26, 4N27, 4N28

Vishay Semiconductors

Optocoupler, Phototransistor Output, with Base Connection

21842



FEATURES

- Isolation test voltage 5000 V_{RMS}
- Interfaces with common logic families
- Input-output coupling capacitance < 0.5 pF
- Industry standard dual-in-line 6 pin package
- Compliant to RoHS directive 2002/95/EC and in accordance to WEEE 2002/96/EC

RoHS COMPLIANT

APPLICATIONS

- AC mains detection
- Reed relay driving
- Switch mode power supply feedback
- Telephone ring detection
- Logic ground isolation
- Logic coupling with high frequency noise rejection

AGENCY APPROVALS

- UL1577, file no. E52744
- BSI: EN 60065:2002, EN 60950:2000
- FIMKO: EN 60950, EN 60065, EN 60335

DESCRIPTION

The 4N25 family is an industry standard single channel phototransistor coupler. This family includes the 4N25, 4N26, 4N27, 4N28. Each optocoupler consists of gallium arsenide infrared LED and a silicon NPN phototransistor.

ORDER INFORMATION	
PART	**REMARKS**
4N25	CTR > 20 %, DIP-6
4N26	CTR > 20 %, DIP-6
4N27	CTR > 10 %, DIP-6
4N28	CTR > 10 %, DIP-6

ABSOLUTE MAXIMUM RATINGS (1)				
PARAMETER	**TEST CONDITION**	**SYMBOL**	**VALUE**	**UNIT**
INPUT				
Reverse voltage		V_R	5	V
Forward current		I_F	60	mA
Surge current	t ≤ 10 µs	I_{FSM}	3	A
Power dissipation		P_{diss}	100	mW
OUTPUT				
Collector emitter breakdown voltage		V_{CEO}	70	V
Emitter base breakdown voltage		V_{EBO}	7	V
Collector current		I_C	50	mA
	t ≤ 1 ms	I_C	100	mA
Power dissipation		P_{diss}	150	mW

For technical questions, contact: optocoupleranswers@vishay.com

Document Number: 83725
Rev. 1.8, 07-Jan-10

Liite 2. Lopullinen ohjelmakoodi.

```
#include "Ubidots.h"

const char* ssid = "...";              // Wifi to connect
const char* pw = "...";                // Password for wifi
const char* UBIDOTS_TOKEN = "...";
Ubidots ubidots(UBIDOTS_TOKEN, UBI_HTTP);

int pinState = 1;
int pin = 4;
int cloudState = 1;

void setup() {
  pinMode(pin, INPUT);
  Serial.begin(115200);
  delay(10);
  ubidots.wifiConnect(ssid, pw);
  ubidots.setDebug(true);
  delay(10);

}

void loop() {
  pinState = digitalRead(pin);
  if (pinState == 1 && cloudState == 0) {
```

```
    float value = 0;
    ubidots.add("OnOff", value);
    bool bufferSent = false;
    bufferSent = ubidots.send();

    if (bufferSent) {
      Serial.println("Value sent");
      cloudState = 1;
    }

  } else if (pinState == 0 && cloudState == 1) {
    float value = 1;
    ubidots.add("OnOff", value);

    bool bufferSent = false;
    bufferSent = ubidots.send();

    if (bufferSent) {
      Serial.println("Value sent");
      cloudState = 0;
    }
  }
  delay(1000);
}
```