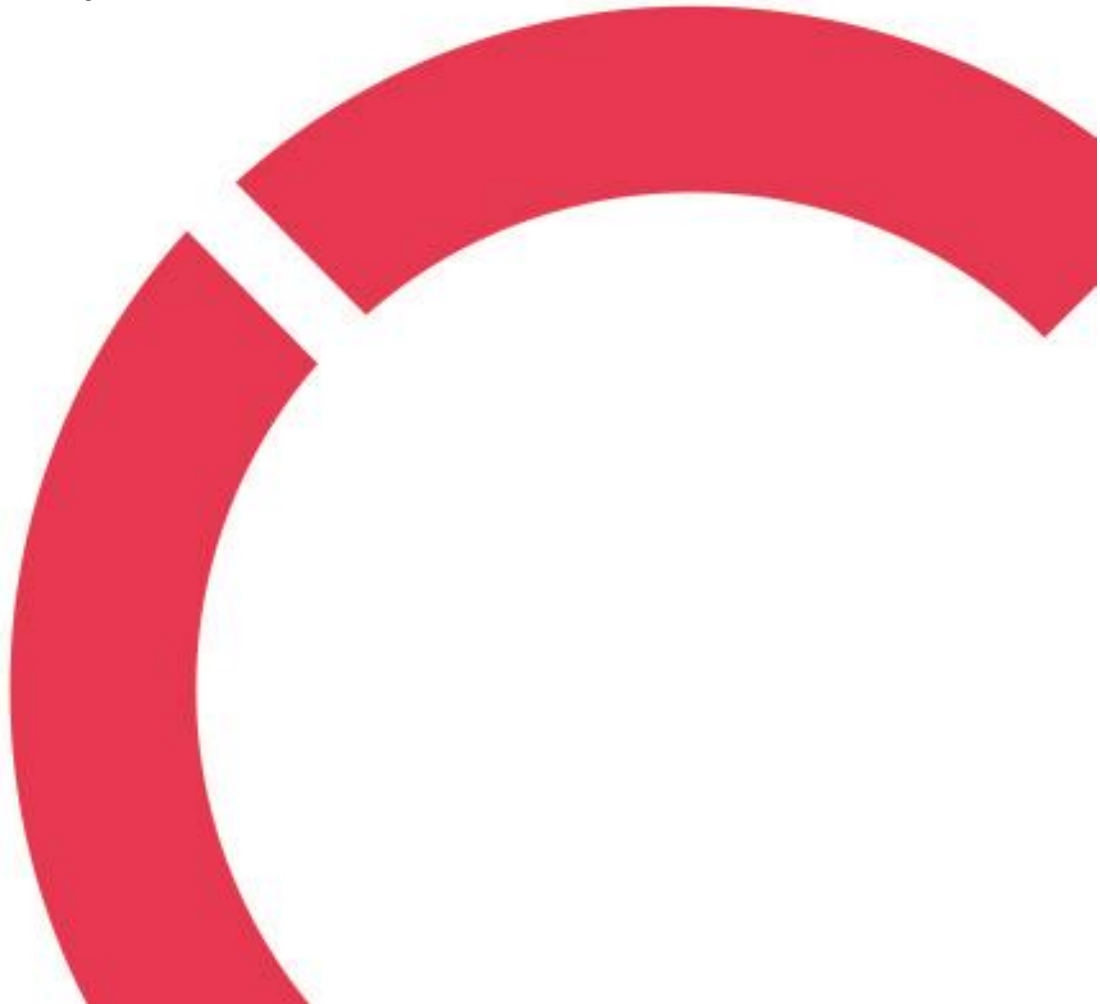


Mehmet Batman
VÄRITUNNISTIN ARDUINO-ALUSTALLA

Opinnäytetyö
CENTRIA-AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma
Toukokuu 2022



TIIVISTELMÄ OPINNÄYTETYÖSTÄ

Centria-ammattikorkeakoulu	Aika Toukokuu 2022	Tekijä/tekijät Mehmet Batman
Koulutus Tietotekniikan koulutusohjelma		<input checked="" type="checkbox"/> AMK <input type="checkbox"/> YAMK
Työn nimi VÄRITUNNISTIN ARDUINO-ALUSTALLA		
Työn ohjaaja Jari Isohanni		Sivumäärä 30 + 8
<p>Tämän opinnäytetyön aiheena on Arduino-alustalle tehty värintunnistin. Opinnäytetyössä käytettiin Arduino-alustaan kuuluvia komponentteja: MKRFOX 1200_-laitetta, Arduino TCS3200_-värianturia ja RGB-LED-valoa. Opinnäytetyö käsittelee näiden kytkentää sekä ohjelmointia.</p> <p>Opinnäytetyön tavoitteena oli tehdä yksinkertainen ja helposti ymmärrettävä toteutus, jotta kuka tahansa perusohjelmoinnin taitava osaa tehdä vastaavan toteutuksen. Ohjelmointi tehtiin Arduino IDE:ssä C++-ohjelmointikielellä. Opinnäytetyön kytkentäkaavio suunniteltiin ja tehtiin Fritzing-ohjelmalla.</p> <p>Opinnäytetyö aloitettiin kytkemällä ja testaamalla RGB-lediä. Tämän jälkeen edettiin värintunnistimen testaamiseen. Työssä huomattiin, että VCC-nastan virta ei riitä värianturille, vaan tuli käyttää suuremman jännitteen 5V-nastaa. Opinnäytteen toteutuksessa oli myös haasteita Sigfox-kirjaston sekä Arduino-IDEn yhteensopivuudessa.</p>		
Asiasanat Arduino, GY-31, MKRFOX1200, RGB, RGB LED, Sigfox, TCS3200, Väri, Värintunnistin.		

ABSTRACT

Centria University of Applied Sciences	Date May 2022	Author Mehmet Batman
Degree programme Information technology		
Name of thesis Arduino based color sensor		
Centria supervisor Jari Isohanni		Pages 30 + 8
Instructor representing commissioning institution or company		
<p>The topic of the thesis was a colour sensor project on the Arduino platform. The thesis used components of the Arduino platform: the MKRFOX 1200 device, the Arduino TCS3200 colour sensor and the RGB LED light.</p> <p>The aim of the thesis was to make simple and easy to understand solution so that anyone with basic programming skills can replicate it. The programming was done in the Arduino IDE using a C++ programming language. The wiring diagram of the thesis was designed and made with Fritzing software.</p> <p>The thesis started by connecting and testing the RGB LED. To test the colour sensor, a VCC pin was first used, but it did not have enough current, so it was changed to a 5V pin. There were also problems with the Sigfox -library which caused problems with the serial monitor in Arduino-IDE.</p>		
<p>Key words Arduino, Color, Color sensor, GY-31, MKRFOX1200, RGB, RGB LED, Sigfox, TCS3200.</p>		

KÄSITTEIDEN MÄÄRITTELY

GND	Jännitteen 0-taso (maa)
LED	Light Emitting Diode, valodiodi
RGB	Punainen (R), vihreä (G), sininen (B) värit
Sketch	Luonnos/Ohjelma
VCC	Käyttöjännite

TIIVISTELMÄ
ABSTRACT
KÄSITTEIDEN MÄÄRITTELY
SISÄLLYS

1 JOHDANTO	1
2 ARDUINO	2
2.1 Arduinon perusteet	2
2.2 Arduino-laitteisto	3
2.3 Arduino ohjelmisto.....	4
2.3.1 Arduino IDE:n napit.....	5
2.3.2 Arduino-IDE valikkopalkki	7
2.3.3 Arduino-laitteiston ohjelmointi.....	10
2.4 MKRFOX 1200.....	14
3 SIGFOX	17
4 VÄRI	18
4.1 Värin aallonpituus.....	19
4.2 Väritunnistin moduuli	21
5 KOEJÄRJESTELYT	24
5.1 RGB-led testi.....	24
5.2 TCS3200(GY-31) värianturi ja RGB led testi	26
6 JOHTOPÄÄTÖKSET	29
LÄHTEET	29
LIITTEET	
KUVIOT	
KUVIO 1. GY-31 Moduuli (TCS3200 Color Sensor Module)	21
KUVIO 2. Funktoni Diagrammi (TCS3200 Block Diagram)	22
KUVIO 3. RGB-led koe piiri ja RGB värit	23
KUVIO 4. RGB-led ja väritunnustuin kytkentäkaavio	25
KUVAT	
KUVA 1. Arduino Uno Piirilevy	4
KUVA 2. Arduino-IDE.....	5
KUVA 3. Arduino-IDE varmista nappi	6
KUVA 4. Arduino-IDE lähete nappi	6
KUVA 5. Arduino-IDE uusi appi	6
KUVA 6. Arduino-IDE avaa nappi.....	6
KUVA 7. Arduino-IDE tallenna nappi	6
KUVA 8. Arduino-IDE sarjamonitori nappi.....	7
KUVA 9. MKRFOX1200	14
KUVA 10. Värispektri	17
KUVA 11. RGB ja CMY väriyhmät	18
KUVA 12. Valonaalto.....	19

KUVA 13. GY-31 Moduuli ja TCS 3200 väritunnistin.....	20
---	----

TAULUKOT

TAULUKKO 1. MKR FOX 1200 Tech Specs	15
TAULUKKO 2. Color vs. Wavelength Range	19

1 JOHDANTO

Valitsin Arduino-alustalle tehdyn väritunnistimen opinnäytetyön aiheeksi, koska olen kiinnostunut Arduino-alustasta ja sillä tehdyistä projekteista. Olen osallistunut ensimmäisen vuoden harjoittelun yhteydessä Arduino-muistipeliin ja Arduino-autojen pieneen projektiin. Ensimmäisten Arduino-projektien tekeminen oli hyvin vaikeaa, koska koodauksen tekemiseksi on tarpeen tietää Arduino-yrityksen kehittämän Arduino IDE-ohjelman käytöstä sekä ohjelmoinnista enemmän.

Joitain valmiita projekteja voidaan käyttää suoraan Arduino IDE:stä, mutta näiden projektien muokkaamiseen tarvitaan perusohjelmointiosaamista ja elektroniikan perustietoja. Kielen ja elektroniikan perustietojen oppiminen vie paljon aikaa. Koska ohjelmointikielissä on laajasti ominaisuuksia, ei tarvitse tietää kaikkia ohjelmointikielen ominaisuuksia Arduino-koodausta varten. Siksi ohjelmointikielen ominaisuudet, joita käytetään usein tässä opinnäytetyössä, selitetään yksinkertaisella kielellä.

Opinnäytetyössä käytetään Arduino MKR FOX 1200 -laitetta, Arduino TCS3200 -värianturia ja RGB-perus-LED-valoa. Tavoitteena on tunnistaa tietyn värin osavärit Arduino-värianturilla, syyttää havaittu väri RGB:ssä ja lähettää viimeinen havaittu väri Sigfox-palveluun. Yksi tavoitteista on, että tämän projektin lopussa jokainen voi helposti tehdä tämän projektin itse.

2 ARDUINO

Tässä luvussa esitellään ensin Arduinon perustoiminnot. Sen jälkeen kerrotaan Arduino-laitteista. Lopuksi kuvataan niiden ohjelmointia. Arduino-laitteiston ohjelmointi myös esitetään tässä kappaleessa.

2.1 Arduinon perusteet

Arduino on avoimen lähdekoodin alusta, johon kuuluvat laitteistot ja ohjelmistot. Arduino-laitetta voi sanoa pieneksi tietokoneeksi, koska tämä järjestelmä sisältää mikrokontrollerin, RAM ja EEPROM –muistit sekä tulo- ja lähtöportteja. Arduinot ovat sulautettuja järjestelmiä ja kerran ohjelmoitu järjestelmä toimii samalla toiminnallisuudella, kunnes se ohjelmoidaan uudelleen. Arduino käyttää yleensä Atmel- tai AVR-prosessoreita korteissaan. Arduinon elektronisten tulo- ja lähtöporttien avulla vastaanotetaan signaaleja piirilevylle ja annetaan signaaleja. Esimerkiksi Arduino voi lähettää signaalikomentoja lampun sytyttämiseksi tai signaalin voi aktivoida painikkeesta. Servomootoria voidaan pyörittää myötä- tai vastapäivään, jolloin moottorille saadaan täydet tai osittaiset tehot. (Arduino. 2018.What is Arduino?.)

Wi-Fi- ja Ethernet-moduulien ansiosta voidaan muodostaa internet-yhteys ja lähettää tai vastaanottaa tietoa. Mittauksia voidaan tehdä Arduinon anturimoduuleilla, esimerkiksi ilman lämpötila voidaan mitata. Elektronisia piirielementtejä käytetään laajasti muodostamaan monimutkaisempia laitteita. Näitä elementtejä ovat mm. vastukset, kondensaattorit, releet, LCD-näytöt ja ajurit. Lampputyypeistä riippuen käytetään erilaisia vastuksia, koska ne suojaavat niitä korkeilta sähkövirroilta. Arduino-levyjen sähköenergia voidaan toteuttaa paristojen tai verkkolaitteen avulla. (Arduino. 2018.What is Arduino?.)

Arduino on suunniteltu kaikkien käyttöön, eikä projektien tekemiseen tarvita insinöörikoulutusta. Arduinon perustajat tekivät Arduinosta kaikin puolin avoimen, lähdekoodi ja laitteisto mukaan lukien. Arduinon perustajat ovat Massimo Banzi, David Cuartielles, Tom Igoe ja David Mellis. Heidän tavoitteenaan on jakaa ja luoda tietoa koodauksesta sekä elektroniikasta. Aiemmin sähköisten projektien tekeminen oli melko kallista ja vei paljon aikaa ja Arduino säästi aikaa ja kustannuksia. (Arduino. 2021.Our Mission & Vision.)

Alkuaikoina Arduinon käyttäjät keskittyivät yleensä viestien lähettämiseen Twitteriin. Suurempia projekteja alettiin tehdä myöhemmin. Erityisesti opiskelijat ovat tehneet onnistuneita robottiprojekteja ja

Japanissa on tehty jopa maanjäristyksen mittauslaitteprojekteja. Arduino-laitteiden laitesuunnitelmat ovat myös avoimia, ja jotkut maat ovat tehneet omat Arduino-laitteensa. Nykyään on tuhansia yrityksiä, jotka myyvät Arduino-laitteita. (Arduino. 2021. Our Mission & Vision.)

2.2 Arduino-laitteisto

Arduinossa on paljon laajennusmahdollisuuksia. Nämä laajennukset koostuvat sarjoista, moduuleista, korteista ja muista pienistä elektronisista elementeistä. Arduino jakaa peruslaitteet kolmeen eri ryhmään. Nämä ovat Classic_(klassinen perhe), Nano_(Nano-perhe) ja MKR_(MKR-perhe). Lisäksi on Arduino Pro -laitteisto ammattilaisille käyttäjille. Arduino suosittelee käyttämään lähtötasolle tehtyjä tuotteita. Esimerkiksi tälle tasolle sopivin setti on aloituspakkaus (starter kit) joka on helppokäyttöinen, ja sopii elektroniikan oppimiseen ja koodaukseen.

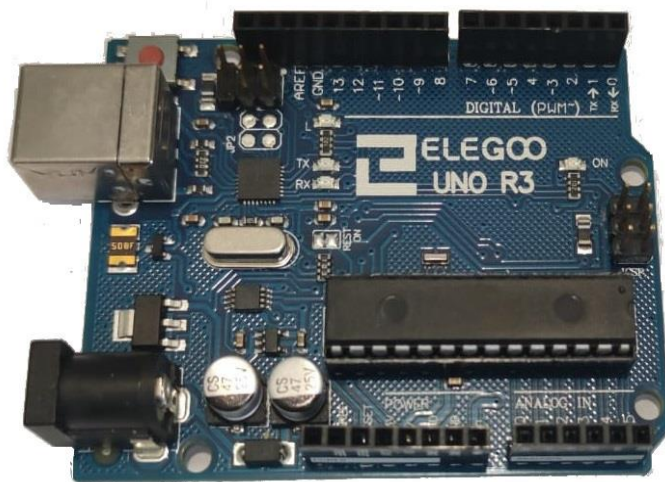
Aloituspakkauksessa on paljon valmiita projektikoodeja. Aloituspakkausten sisältö riippuu hieman myyjästä. Esimerkiksi Arduinon verkkosivustolla myydyin Starter-Kitin aloitussarja sisältää projektikirjan, servomoottorin, tavallisen moottorin, LCD-näytön, Arduino kytkentälevyn 400 pinniä, kondensaattorin, vastuksen, painikkeen, summerin, vastuksen, lämpötila-anturin, diodin, USB-kaapelin, akun liitäntäkaapelin, punaisen LEDin, keltaisen LEDin, vihreän LEDin, sinisen LEDin, RGB-LEDin ja Arduino Uno -laitteen. Tämän pakkauksen hinta (2022) on 79,90 €. (Arduino. 2022. Hardware; Arduino Starter Kit Multi-language 2021.)

Tässä työssä käytettävässä Starter-sarjassa on lisäksi RFID-moduuli, äänitunnistin, 7 segmentin näyttö, kosteusmittari, IR-moduuli, näppäimistömoduuli, virtalähde, rele, kellomoduuli ajankäyttöön, joystick-moduuli(ohjaussauva), askelmoottori, liiketunnistin, 830 pinnin kytkentälevy ja ultraäänianturi. Sarjan hinta on noin 54,50 € Amazonissa (toukokuu 2022). (Amazon.com 2022; ELEGOO UNO R3 Project Most Complete Starter Kit 2022.)

Arduinolla on olemassa erityisesti laitteita, joissa on valmis verkkoyhteysmoduuli. Muun muassa MKR-sarjan korteissa on valmiina erilaisia moduuleja. Esimerkiksi Arduino MKR 1000 WIFI -kortilla on valmiina WIFI-moduuli. Lisäksi on saatavilla wan-, sim-, tai sigfox-moduuleja. Nano-sarja on pienempi kuin tavallinen Arduino-laite. Se voidaan yhdistää wifi-moduulin tai bluetooth:n kautta. Tiedonvaihto on erittäin helppoa. (Arduino. 2022.Products. Entry Level.)

Arduinon käyttöönottoa helpottavat esim. Educatin(koulutus) pakkaukset, jotka ovat saatavilla opiskelijoille ja opettajille. Educat tarjoaa jokaiselle koulutuksen tasolle omia tuotteitaan. Esimerkiksi on teknisiä sarjoja, luokkasarjoja, fysiikan laboratoriosarjoja ja muita sarjoja.

Kuvassa on klassinen sarja perus Arduino-laitteisto. Siinä on ATmega328P-prosessi, kuvassa 1 on esitetty sarjan Arduino ELEGOO UNO laite. Se on yleisimmin käytetty kortti yksinkertaisille projekteille. (Arduino. 2022.Products. Entry Level)



KUVA 1. Arduino ELEGOO Uno Piirilevy 2022

2.3 Arduino ohjelmisto

Arduino-laitteistoa ohjataan ohjelmointikielen kautta, ohjelmat kirjoitetaan Arduino-IDE:ssä. Se myös yhdistää tietokoneen ja Arduino-laitteen. Arduino-IDE toimii Windows-, Linux- ja Mac-käyttöjärjestelmissä. Se on avoimen lähdekoodin toteutus, jota voidaan käyttää kaikille Arduino-korteille. Nykyinen versio on Arduino-IDE 1.8.19, mutta tuleva versio IDE 2.0 RC on myös ladattavissa. Arduinon Github-sivu sisältää koodeja ja kaikki Arduinoon liittyvä dokumentaatio löytyy sieltä. Kuvassa 2 on Arduino-IDE , jota käytettiin työssä. (Overview of the Arduino IDE 1 2022.)

Kirjoitetut ohjelmat tallennetaan .ino tiedostotunnisteella tallennettaessa. Editorissa on mm. kopioi-, leikkaa-, liitä-, tekstihaku- ja korvausominaisuuksia. Kun Arduino-laite yhdistetään tietokoneeseen editorin alaosassa näkyy tietoja laitteesta ja sen tilasta. Kuvassa 2 näkyy yhdistettynä Arduino-laite (Arduino MKRFOX 1200) portissa Com 3. Arduino-IDE:ssä on neljä osaa, nämä ovat valikkorivi, painikepalkki, koodausalue ja viestiosa. Valkoinen kenttä on koodausosa, ja ohjelman koodi kirjoitetaan siihen. (Overview of the Arduino IDE 1 2022.)

```

sigfoxtest | Arduino 1.8.16
Tiedosto Muokkaa Sketsi Työkalut Apua

sigfoxtest
#include <SigFox.h>
#include <ArduinoLowPower.h>

void setup() {

  Serial.begin(9600);
  SigFox.begin();

  String version = SigFox.SigVersion();

  String ID = SigFox.ID();

  String PAC = SigFox.PAC();

  // Display module information

  Serial.println("MKRFox1200 Sigfox first configuration");

  Serial.println("SigFox FW version " + version);

  Serial.println("ID = " + ID);

```

24 Arduino MKR FOX 1200 on COM3

KUVA 2. Arduino-IDE 2022

2.3.1 Arduino IDE:n napit

Tässä osiossa esitellään Arduino-IDE:n peruspainikkeet, joita käytetään hyvin usein, ne ovat Tarkista, Lähetä, Uusi, Avaa, Tallenna, Serial-monitori (sarjaportti seuranta).

Tarkistaa koodit (KUVA 3), kääntää koodin ja antaa virheilmoituksen yllä olevan kuvan 2 mustalle alueella, jos löytyy koodausvirheitä. Tämän annetun virhekoodin avulla voidaan löytää kohta, joka korjataan



KUVA 3. Tarkista-nappi

Lähetä koodit -nappi (KUVA 4.): Arduino-IDE lähettää koodausosan koodit Arduino-laitteeseen yhteysportin kautta. Tämä painike lähettää järjestelmään vain tarkistettuja koodeja. Jos koodeissa on virheitä, se lähettää virheilmoituksen.



KUVA 4. Lähetä-nappi

Uusi-ohjelma (KUVA 5.), avaa uuden tyhjän ohjelman. IDE:ssä on mahdollista avata niin monta uutta ohjelmaa kuin halutaan. Uudet sivut avautuvat uuteen ikkunaan, eivätkä ne vaikuta käytettävään projektiin.



KUVA 5. Uusi-nappi

Avaa-Napin (KUVA 6.), avulla voidaan avata tallennettu ohjelmia tietokoneen muistiin muistikorteille tai USB-laitteille.



KUVA 6. Avaa-nappi

Tallenna-Napin (KUVA 7.) avulla voidaan tallentaa ohjelma.ino-päätteellä.



KUVA 7. Tallenna-nappi

Sarjamonitoria (KUVA 8.) käytetään Arduino-laiteesta lähetettyjen tietojen näyttämiseen. Nämä liitännät voidaan tehdä USB- tai COM-liitäntäkaapeleilla. Sarjamonitorin nopeus vaihtelee projektista riippuen, nopeus voidaan asettaa

```
Serial.begin (9600);
```

-komennolla koodissa, jossa 9600 on sarjaportin nopeus.



KUVA 8. Sarjamonitori -nappi

2.3.2 Arduino-IDE valikkopalkki

Tässä luvussa esitellään Arduino-IDE:n työkalun valikkopalkin komennot. Arduino-editorissa on seuraavat valikot: **Tiedosto**, **Muokkaa**, **Ohjelma** ja **Työkalut**. Nämä valikot esitellään seuraavissa kappaleissa.

Tiedosto

Arduino IDE:n osa, jossa perustoiminnot sijaitsevat. Nämä painikkeet ovat **Uusi**, **Avaa**, **Avaa viimeksi**, **Luonnoskirja**, **Sulje**, **Tallenna**, **Tallenna nimellä**, **Sivun asetukset**, **Tulosta**, **Asetukset** ja **Lopeta**. Painikkeiden toimintaa kuvataan lyhyesti alla.

Uusi: Avaa uuden tyhjän ohjelman, peruskoodi on valmiina avautuvalla sivulla.

Avaa: Avaa aiemmin tallennetut ohjelmat tietokoneelta tai muulta tallennusvälineeltä.

Avaa viimeksi: Näyttää viimeksi käytetyt ohjelmat.

Luonnoskirja (Sketchbook): Näyttää Arduino-kansioon tallennetut ohjelmat tietokoneen dokumentaatiokansiosta. Arduinon asennuksen mukana tulee valmiita ohjelmia, jotka löytyvät täältä valmiiksi.

Sulje: Sulkee avoimen ohjelman. Jos koodiin on tehty muutoksia ja painetaan painiketta, saadaan viesti, jossa kysytään, halutaanko koodien lopullinen versio tallentaa.

Tallenna: Tallentaa ohjelman nykyisellä nimellä.

Tallenna nimellä: Ohjelma tallennetaan eri nimellä kuin nykyinen nimi. Tämä tehdään yleensä silloin, kun halutaan säilyttää alkuperäinen tiedosto.

Sivun asetukset: Määrittää tulostimesta tulostettavien koodien sivukoot.

Tulosta: Käytetään ohjelman tulostamiseen (PDF tai paperi).

Asetukset: Arduino IDE:n omat asetukset voidaan tehdä täällä. Esimerkiksi oletusasetuksissa koodin Näytä rivinumero -osaa ei ole valittuna. Tätä asetusta voidaan käyttää, jos halutaan näyttää koodien rivinumerot. Myös kansio, johon luonnos tallennetaan, voidaan vaihtaa täältä.

Lopeta: Sulkee koko IDE:n.

(Arduino Integrated Development Environment (IDE) v1 2022.)

Muokkaa

Tässä valikossa ovat muokkausnapit, joita käytetään usein tekstiä tai koodia kirjoitettaessa. Nappeja on esimerkiksi: **Valitse kaikki**, **Leikkaa**, **Kopioi**, **Liitä**, **Kumoa** ja **Kopioi**. Napit ovat tuttuja myös muista tekstinkäsittelyohjelmista.

Kumoa: Käytetään viimeisimmän toiminnon kumoamiseen.

Leikkaa: Leikkaa (poistaa) valitun tekstin ja kopioi sen leikepöydän muistiin.

Kopio: Kopioi valitun tekstin leikepöydän muistiin.

Kopio forumille: Kopioi valitun tekstin leikepöydän muistiin, foorumeille sopivassa muodossa.

Kopioi HTML:nä: Kopioi valitun tekstin leikepöydän muistiin HTML-muodossa.

Liitä: Lisää leikepöydän muistissa olevan tekstin editorin tekstialueeseen.

Valitse kaikki: Valitsee koko tekstialueen

Go to line: Siirtyy halutulle riville.

Kommentointi päälle/pois: Tekee valituista riveistä kommenttitekstiä (tai poistaa kommentin)

Lisää sisennystä: Lisää sisennystä tekstiin.

Vähennä sisennystä: Vähentää sisennystä tekstissä.

Etsi: Etsii tietoja tekstistä.

Etsi seuraava/edellinen: Etsii seuraavan kohdan tekstistä.

(Arduino Integrated Development Environment (IDE) v1 2022.)

Ohjelma

Ohjelma-valikon kautta suoritetaan komentoja, jotka liittyvät avoinna olevaan ohjelmaan, sekä yhdistettyyn Arduino-laitteeseen.

Vahvista/Käännä: Tarkistaa koodin ja kääntää sitten koodin binäärikoodiksi

Lähetä: Tarkistaa koodin ja kääntää sitten koodin binäärikoodiksi sekä lähettää käännetyn koodin Arduino-laitteelle yhteysportin kautta. Liitäntäkaapeli voi olla USB- tai RS232-liitäntäkaapeli.

Lähetä ohjelmointilaitteella: Tätä ominaisuutta varten on ensin aktivoitava ”polta käynnistyslataaja” asetukset-valikossa. Se koodaa Flash-muistin, mikä on tarkoitettu ammattimaisiin toimintoihin ja voi vahingoittaa Arduino-korttiasi.

Vie käännetty binaari: Arkistoi koodin käännettynä .hex tiedostona.

Näytä sketsikansio: Avaa käytetyn ohjelmakansion.

Sisällytä Kirjasto: Avaa osion, jossa asennetaan kirjastoja. Jos tarvitsee lisätä kirjasto, se voidaan liittää .zip tiedostona IDE:n järjestelmänvalvojat-osiossa.

Lisää tiedosto: Lisätään ohjelmakoodia Arduino IDE-projektiin.

(Arduino Integrated Development Environment (IDE) v1 2022.)

Työkalut

Arduino IDE:n neljäs valikko-osa on työkalut. Tässä osassa on erilaisia työkaluja niin laitteen kuin koodin hallintaan.

Arkistoi ohjelma: Tallentaa ohjelman .zip tiedostona samaan kansioon kuin sketsi.

Korjaa koodaus ja avaa uudelleen: Peruuttaa koodiin tehdyt muutokset ja avaa koodin uudelleen.

Ylläpidä kirjastoja: Avautuu kirjastojen hallintasivu, jossa voidaan ladata kirjastoja projektiin internetin kautta.

Sarjamonitori: Avaa Sarjamonitori-sivun ja näyttää portin kautta vastaanotetut tiedot. Voidaan myös lähettää tietoja laitteelle, esimerkiksi kirjoittaa toiminnot, jotka halutaan tulostaa näytölle sarjanäyttöön.

Sarjapiirturi: Suorittaa saman tehtävän kuin Sarjamonitori, mutta antaa tuloksen graafisesti.

Wifi-päivitys: Jos käytetyssä Arduino-kortissa on Wifi, voidaan päivittää laite Wifin kautta.

Laite: Näyttää laitteen, johon halutaan muodostaa yhteys. Arduinon laitoksella on tässä valmis kortti-luettelo. Jos käytettävää korttia ei ole luettelossa, kortinhallinta ladataan Internetiin ja Arduino IDE:n.

Hae laitteen tiedot: Näyttää laitteen nimen, sen sarjanumeron, vid-numeron ja pid-numeron.

Ohjelmoija: Tässä käytettävät ohjelmoijat laitteelle. Normaalisti tätä ei tarvitse käyttää, koska Arduino tekee valinnan automaattisesti. Jos kortin siru on vaihtunut, tämän kautta voidaan valita oikea siru.

Polta käynnistyslaataja: Tarvitaan, kun käytetään uutta mikrokontrolleria. Tämä ei ole välttämätöntä normaaliin käyttöön, koska voi vahingoittaa Arduino-korttia.

(Arduino Integrated Development Environment (IDE) v1 2022.)

Apua (Ohje)

Tämä valikko sisältää ohjedokumentaatioita. Ohjeet ovat kaikkien saatavilla aloittelijasta ammatilliseen tasoon.

Aloittaminen: Kaikki tarvittavat tiedot Arduinon käytön oppimiseen löytyvät tältä sivulta.

Esimerkiksi käyttäjää Windows-, Linux- ja MacOS-käyttöjärjestelmissä ohjeistetaan askel askeleelta. Lisäksi tietoa Arduino-verkkoeditorin käytöstä on saatavilla.

Ympäristö: Arduino IDE: tietoa sen käytöstä.

Vianmääritys: Tämä valikko johtaa Arduinon ohjekeskuksen sivulle.

Opas: Täältä löytyy ohjeita Arduino IDE:n käyttämiseen sekä ohjelmointiin

Etsi oppaasta: Täältä voi etsiä tietoa oppaasta

Usein kysytyt kysymykset: Vastauksia usein kysytyihin kysymyksiin

Vieraile Arduino.cc: Avaa Arduinon virallisen verkkosivun selaimen.

(Arduino Integrated Development Environment (IDE) v1 2022.)

2.3.3 Arduino-laitteiston ohjelmointi

Arduino-laitteiden ohjelmointiin tarvitaan ohjelmointikieltä. Laitteesta riippuen voidaan käyttää C- tai C++-ohjelmointikielen koodeja Arduino-IDE:n kanssa. Arduinoa ohjelmoitaessa käytetään kielten perusrakenteita kuten muuttujia, toimintoja ja rakenteita.

Arduino-laitteisiin yhdistettäviä komponentteja voidaan ohjata nastojen kautta. Nastoissa on INPUT- ja OUTPUT-komennot, jotka asettavat tulon ja lukevat sen. Kirjoitettaessa HIGH-/ LOW-nastat asetetaan ohjelmallisesti joko todeksi ja ei-todeksi. Tämä on yksi Arduinon tärkeimmistä ominaisuuksista. (Arduino. 2020. Language Reference.)

Seuraavissa kappaleissa on kuvattu tärkeimpiä funktioita ohjelmoitaessa Arduino-laitteita.

digitalWrite() -funktioita käytetään nasta-arvojen HIGH ja LOW asettamiseen. Tällä toiminnolla on kaksi parametria. Nämä ovat käytettävän nastan nimi ja arvo. HIGH-arvo vaihtelee korttien mukaan, se voi olla 5V tai 3.3V. LOW-parametria käytetään määrittämään arvo 0 v. Tämä funktio ei palauta arvoa. (Arduino. 2020. Language Reference.)

Funktiota siis käytetään:

```
digitalWrite(nastan-nimi tai -numero, arvo)
```

Esimerkki funktion käytöstä:

```
digitalWrite(3, HIGH); // Nasta 3:lle toimitetaan 5V virta .
digitalWrite(4, LOW); // Nasta 4:lle toimitetaan 0V virta.
```

analogWrite () -funktiolla voidaan antaa analogisia arvoja (pwm-aaltoja) Arduino-nastoille. Tämän toiminnon ansiosta voidaan esim led-lampun kirkkautta hallita. Funktiota käytetään myös Arduino-moottoreiden nopeustason säätämiseen. (Arduino. 2020. Language Reference.)

Funktiota siis käytetään:

```
analogWrite(nastan numero tai nastan nimi, arvo)
```

Arvo määritellään lukuna väliltä 0–255, 0 (LOW) ja 255 (HIGH).

Esimerkki funktion käytöstä:

```
analogWrite(5, 255); // Nastalle 5 toimitetaan 5V energiaa.
analogWrite(6, LOW) // Nastalle 6 toimitetaan 0V energiaa.
```

pinMode() -Funktiota käytetään Arduino-laitteen nastojen tulo- tai lähtönastan toiminnan asettamisen joko tulo- tai lähtötyypiksi. Funktiolla on kaksi parametria, nastan nimi tai numero ja asetuksen tyyppi. Tyypin parametri OUTPUT määrittää nastan ulostuloksi ja INPUT syötöksi. (Arduino. 2020. Language Reference.)

Funktiota siis käytetään:

```
pinMode (nastan-nimi tai -numero, tulo-lähtötila).
```

Esimerkiksi kirjoitetaan koodi punaiselle LED-valolle, asettamalla aluksi nasta ulostuloksi, ja sen jälkeen asetetaan sille arvo HIGH.

```

void setup(){
// Digitaalinen nasta kaksi asennettu (punainen LED) lähtönastana.
    pinMode(2, OUTPUT);
}
void loop(){
// Punaiselle ledille annetaan 5 V:n tai 3,3 V:n virta ja lamppu palaa.
    digitalWrite(2, HIGH);
    delay(500); //Punainen led 500ms (millisekuntia) on palaa.
    // Punainen ledin nastalle annetaan 0 v ja lamppu sammuu.
    digitalWrite(2, LOW);
    delay (500); // Punainen ledi 500ms (millisekuntia) sammuu.
}

```

setup()-funktio

Arduino-koodin funktio, jota kutsutaan kerran ohjelman käynnistyessä. Tässä funktiossa tehdään yleiset asetukset, kuten sarjamonitori-, muuttuja- ja pinmode-asetukset. (Arduino. 2020. Language Reference.)

Esimerkiksi RGB-LED-valon ulostulopinnien asettaminen setup-funktiossa.

```

int rledpin= 2;           // RGB-LED-valon punainen LED-jalka on kytketty nastaan 2.
int gledpin= 3;          //RGB-LED-valon vihreä LED-jalka oli kytketty nastaan 3.
int bledpin=4;           //RGB-LED-valon sininen LED-jalka on kytketty nastaan 4.

void setup(){
pinMode(rledpin,OUTPUT); // Punainen LED-nasta on asetettu lähtöön.
pinMode(gledpin, OUTPUT); //Vihreä LED-nasta on asetettu lähtöön.
pinMode(bledpin, OUTPUT); //Sininen LED-nasta on asetettu lähtöön.
}
void loop(){
digitalWrite(rledpin,HIGH);
}

```

delay() -funktio, käytetään jotta saadaan Arduino-laitteen prosessori odottamaan määritetyn ajan. Funktion parametrina on ms (millisekunti), eli 0.001 sekuntia. Enimmäisarvo parametrilla voi olla unsigned long((2³²)-1~ 4 294 967 295). Seuraavassa esimerkissä punainen LED syttyy 0,5 sekunnin ajaksi ja sammuu jälleen 0,5 sekunnin ajaksi. (Arduino Language Reference, 2020.)

```
int punoitettiin =5;           //Punaisen johdon liittäminen nastaan 5.
void setup(){
pinMode(redled, OUTPUT);      //Punaisen LED-tapin asettaminen lähtöön.
}
void loop(){
digitalWrite(redled,HIGH);     //Punainen ledi saa virtansa 5 V:lla tai 3, 3 V:lla.
delay(500);                    //500ms =0,5 sekuntia pidätys.
digitalWrite(redled,LOW);      //Punainen ledi toimii 0v:llä, ja lamppu sammuu.
delay(500);                    //500ms =0,5 sekunnin pito.
}
```

pulseIn() -funktio suorittaa nastan LOW- ja HIGH-muutosten pulssimuotoisen seurannan. Tämä toiminto käyttää yleensä kahta parametria, mutta sillä voidaan antaa myös kolme parametria. Parametrit ovat nasta, arvo ja aikakatkaisu. (Arduino. 2020. Language Reference.)

```
pulseIn(nasta,arvo)
pulseIn(nasta,arvo,aikakatkaisu)
(Arduino. 2020. Language Reference.)
```

void loop()

Arduinin toiminto, joka suorittaa koodia jatkuvassa silmukassa IDE-ohjelmassa.

```
void loop()
{
//toimii silmukan matolla, jossa nämä koodit pyörivät.
digitalWrite(redled,HIGH);     //Punainen ledi saa virtansa 5 V: lla tai 3, 3 V: lla.
delay(500);                    //500ms =0,5 sekuntia pidätys.
digitalWrite(redled,LOW);      //Punainen led toimii 0v: llä, ja lamppu sammuu.
delay(500);                    //500ms =0,5 sekunnin pito.
}
```

Tämä funktio (tyhjä silmukka) on valmis osa ohjelmaa, kun uusi ohjelma Arduino IDE:ssä avataan. (Arduino. 2020. Language Reference.)

map() funktio suorittaa käännöksen (skaalaus) annetuista numeroalueista toisiin lukualueisiin. Skaalaus suoritetaan annetuista arvoista annettuihin minimi- ja maksimiarvoon.

`map(value, fromLow, fromHigh, toLow, toHigh)`

Parametrina annetaan arvot: minimi ennen kääntämistä, maksimi ennen kääntämistä ja minimi kääntämisen jälkeen funktio palauttaa skaalatun arvon. Esimerkiksi:

```
green = map (green, 83,975, 0, 255);
```

green: käsiteltävä arvo.

83: minimi ennen kääntämistä.

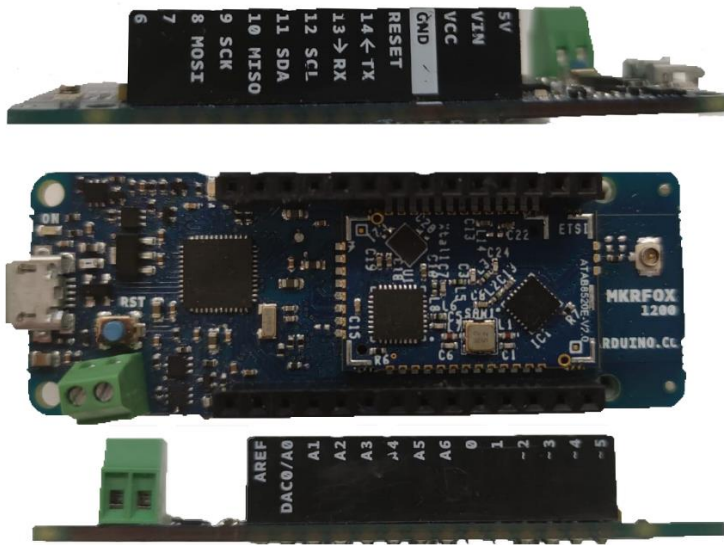
975: maksimi ennen kääntämistä.

0: minimi käännöksen jälkeen.

255: maksimi kääntämisen jälkeen. (Arduino. 2020. Language Reference.)

2.4 MKRFOX 1200

Mkrfox1200 sisältää melkein kaikki tavallisen Arduino-laitteen ominaisuudet. Tämän kortin avulla voidaan kuitenkin helposti muodostaa yhteys Sigfox-verkkoon. Myös laitteen virrankulutus on minimaalinen, koska valmistaja on suunnitellut sen kestäväksi pitkään, erityisesti akun kanssa. Laitteessa on Arm Cortex-M0 prosessori (32-bittinen), lisäksi laitteessa on 'Microchip Smart RF ATA8520' -siru Sigfox-yhteyttä varten. Yleensä Arduino-laitteet toimivat 5 V:n teholla, mutta niille voidaan antaa suurempaa jännitettä. Esimerkiksi Arduino-aloitussarjan paketti sisältää virtalähteen, jossa on 9 V ja 1 A virta. Joskus Arduinon lisämoduulit tarvitsevat ylimääräistä virtaa. Esimerkiksi moottorimoduuli toimii paremmin 9V-jännitteellä. MKRFOX 1200 sen sijaan voi ottaa vastaan vain 5 V jännitteen, se voi toimia myös kahdella 1.5 V:n paristolla (yhteensä 3 V). MKRFOX1200 ei ole oletuksena käytettävissä Arduino IDE:n asetuksissa vaan tähän käyttöön tulee ladata Arduino SAMD -korttien paketti etsimällä tämä laitehallinnasta (Työkalut>Kortit>BoardManager). Alla kuvassa 9 on esitetty Arduino MKRFOX1200 -laite. Seuraavalla sivulla (TAULUKKO 1) on esitetty teknisiä tiedot laitteesta. (Arduino. 2022.MKR FOX 1200.)



KUVA 9. MKRFOX1200 (Batman 2022.)

TAULUKKO 1. (mukaillen Arduino MKR FOX 1200 Tech Specs 2022)

Kortti	Nimi	Arduino MKRFOX 1200
	SKU	ABX000014
	Yhteensopivuus	MKR
Mikro-ohjain	SAMD21 Cortex-M0+32- bittinen pienitehoinen ARM	
USB-liitin	Micro USB (USB-B)	
Pinssit	Digitaaliset I/O-pinnit	8 kpl
	PWN pinnit	13 kpl (0–8,10,12 A3, A4)
	Analogisen tulon pinnit	7(ADC 8/10/12 bitti)
	Analogiset lähtöliittimet	1(DAC 10-bittinen)
	Ulkoiset keskeytykset	10(0,1,4,5,6,7,8,9, A1, A2)
Yhteydet	SigFox	Microchip Smart RF ATA8520
	Toimialue	EU(Eurooppa)
	Kantoaallon taajuus	868 Mhz (Megahertsiä)
	Kellonopeus	32,768 kHz (Kilohertsiä) (RTC),48 MHz (Kilohertsiä)
	Antenni teho	2 dB (Desibeli)

	Kantoaallon taajuus	868 Mhz (Megahertsiä)
Kellonopeus	Proessori	48 MHz
Tehoa	Levyvirtalähde (USB/VIN)	5 V
	Piirin käyttöjännite	3,3 V
	Digitaaliset I/O-pinnit	8 kpl
	DC-virta per I/O Pinni	7 mA(milliampeeri)
	Tuettu akku	2 x AA tai AAA
Muisti	SAMD21G18A	256 kt(kilotavu) Flash, 32 kt(kilotavu) SRAM
Mitat	Pituus	67,64 mm(millimetriä)
	Leveys	25 mm(millimetriä)
	Paino	32 gr(gramma)

3 SIGFOX

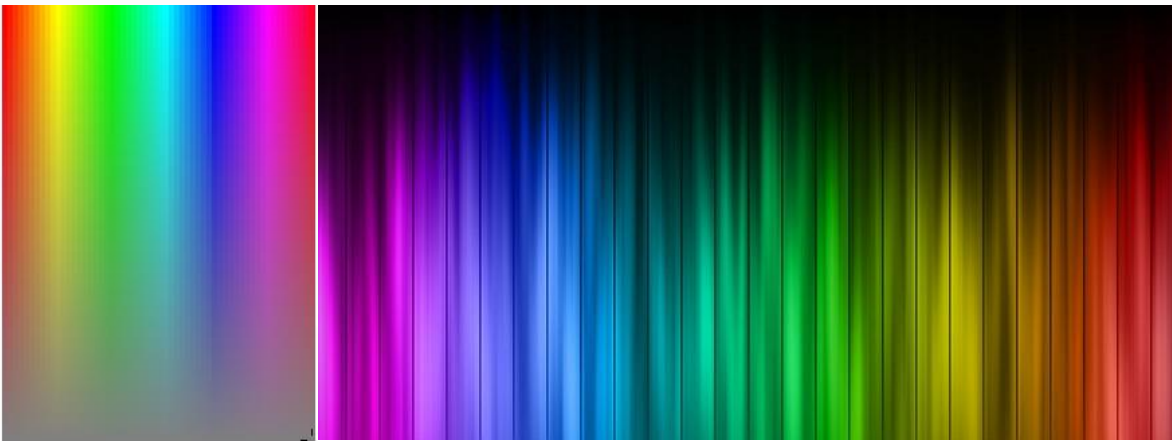
Ludovic Le Moan ja Christophe Fourtet perustivat Sigfoxin vuonna 2010 startup-yrityksenä. Yhtiön perustajat kehittivät Sigfox-järjestelmän, jolla voi lähettää viestejä pitkiä matkoja langattomilla radio-signaaleilla käyttämällä vähän virtaa, pienin kustannuksin. Useita vuosia yhtiön perustamisen jälkeen Moan ja Fourtet saivat rahoitusta, ja yhtiö on kasvanut hyvin nopeasti ja saanut satoja miljoonia euroja investointeja Amerikasta, Euroopasta ja Aasiasta. Sigfox-järjestelmä toimii LPWAN-luokassa, Low Powered Wide Area Network. Vuonna 2017 Sigfox-verkko kattoi 500 miljoonan ihmisen alueen, ja toimi 30 maassa. Nykyään kattavuus on noin 75 maata ja 1,5 miljardia ihmistä. Video- tai ääniviestejä ei ole mahdollista lähettää Sigfox-verkossa, vaan sitä käytetään IoT-laitteiden viestintään. Sigfox-verkko rajoittaakin lähetettyjen viestien kokoa on, ja niiden pituuden on oltava 12 tavua, vastaavasti laite voi vastaanottaa korkeintaan 8 tavun viestejä. Sigfox-verkko rajoittaa myös viestien määrää, päivittäin voidaan lähettää maksimi 140 kappaletta viestejä. (Isohanni 2017; Sigfox Tech Overview and Resources; IoT use cases for agriculture; Our Story.)

Sigfox-verkkoa käytetään mm. maataloudessa, teollisuudessa, kuljetuspalveluissa, postinjakelussa, älykkäissä kotijärjestelmissä ja monilla muilla aloilla. Sigfox-laitteita käytetään esimerkiksi maataloudessa maitotuotteiden säilytystilojen lämpötilan säätämiseen sekä säiliöiden käyttöasteen tarkistamiseen. Lisäksi GPS-moduulin ansiosta voidaan paikantaa eläimiä, tämä lisää kuitenkin virrankulutusta. (Isohanni 2017; Sigfox Tech Overview and Resources; IoT use cases for agriculture; Our Story)

Sigfoxin rajoitusten ansiosta, viestin koko ja päivittäinen raja, laitteen paristot kestävät kauemmin kuin jatkuvan yhteyden langattomissa laitteissa. Verkon kattavuus Suomessa on noin 90 % ja kattavuus Virossa on noin 69 %. Monissa Suomen kaupungeissa on Digfox-tukiasema. Vuonna 2017 Sigfox-järjestelmässä oli noin 20 miljoonaa rekisteröityä laitetta. Kahden viime vuoden aikana viestien määrä on kasvanut 16 miljoonalla 79 miljoonaan. (Sigfox Tech Overview and Resources; IoT use cases for agriculture; Our Story)

4 VÄRI

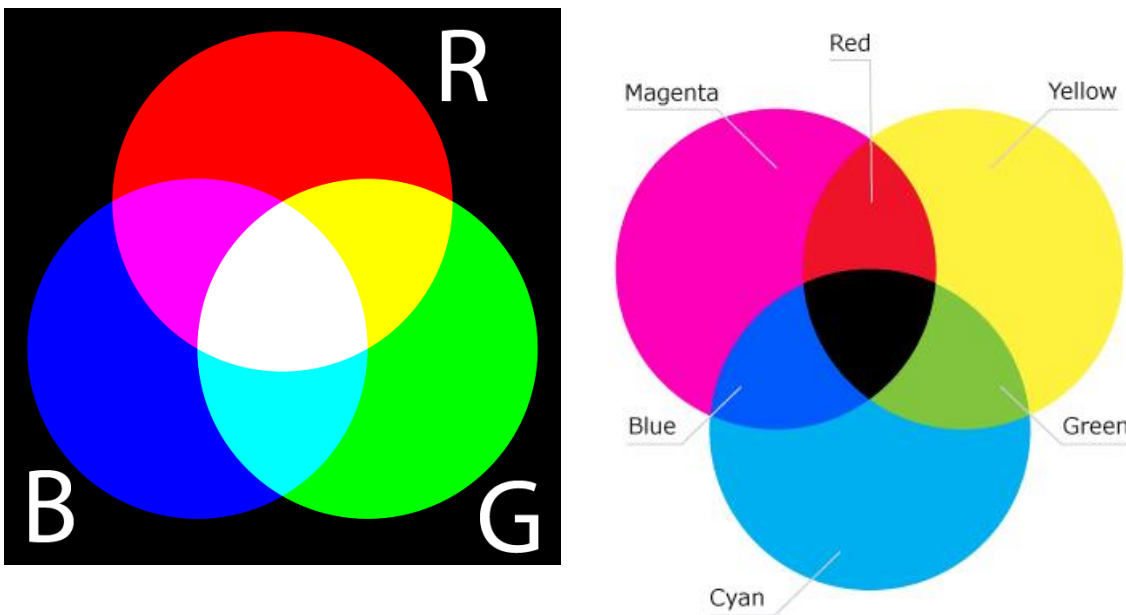
Valkoisen auringonvalon taittuu ja näin siitä voidaan erotella muita värejä, valkoinen väri muodostuu siis useista väreistä. 1700-luvulla tunnettu fyysikko ja matemaatikko Isaac Newton suoritti optisia testejä. Talonsa pimeässä huoneessa hän havaitsi värispektrin kuljettamalla valkoisen auringonvalon ohi reikää. Muut jatkoivat Newtonin tutkimuksia myöhemmin. Värejä käytetään elämämme kaikilla osa-alueilla, kuten fysiikassa, kemiassa, luonnossa, terveydenhuollossa, teollisuudessa. Aiemmin valon uskottiin olevan jumalallinen elementti. Aristoteles omaksui teorian, jonka mukaan värit ovat valkoisia ja mustia. Hänen teoriansa pysyi voimassa yli 1000 vuotta. Kun Newton julkaisi teorian, Aristoteleen teoria mitätöitiin. Newton-kokeessa hän tunnisti seitsemän väriä värispektrissä. Havaitut värit olivat punainen väri, oranssi väri, keltainen väri, vihreä väri, sininen väri, vaaleansininen väri ja violetti väri. Kuvassa 10 on esitetty värispektri. (Newton's Rainbow.)



KUVA 10. Värispektri (muokaten Microsoft Office väripaletti; pixabay Lightshow Matryx)

Kuuluu saksalainen runoilija, kirjailija ja tutkija Goethe keksi teorian, jonka mukaan ihmiset kokevat värit eri tavoin. Hänen teoriansa hyväksyi ja sovelsi laaja joukko taiteilijoita. Muun muassa valkoinen väri koetaan puhtaana, punainen yhdistetään liikkuvuuteen ja musta antaa levottomuutta. Myös yritykset käyttävät värejä, joilla on psykologinen vaikutus ihmisiin mm. tuotteidensa pakkauksissa. Värien käyttö on myös eläimille tärkeää mm. naamioinnin osalta. Televisiot, tietokoneen näytöt, tabletin näytöt ja matkapuhelimen näytöt heijastavat valoa värien luomiseksi. Digitaalisissa järjestelmissä pääväreinä käytetään punaista, vihreää ja sinistä (RGB). Nämä kolme väriä muodostavat yhdessä valkoisen valon. RGB-värejä käytetään myös verkkosivuilla ja muissa digitaalisissa järjestelmissä, digitaalisten logojen suunnittelussa. Tulosteissa taas ensisijaisina väreinä käytetään syaanina, magentaa, keltaista (CMY). Sekoittamalla nämä värit saadaan musta väri. Kolmen värin sekoittaminen tulostuksessa vaatii

kustannuksia ja aikaa, joten erillisenä värinä käytetään puhdasta mustaa. Siksi tulostusvärejä kutsutaan CMYK-väreiksi. (Newton's Rainbow; RGB vs CMYK: What's the difference? 2019.)



KUVA 11. RGB ja CMYK väriyhdyt (mukaan RGB OpenClipart-Vectors; Colour mixing 2022)

:

RGB käytetään sanomalehdissä, televisiolähetysissä ja sosiaalisen median kanavissa, jotka lähettävät Internetissä. RGB-luokitus on käytettävissä kaikilla visuaalisilla alueilla. CMYK:ä käytetään fyysisissä tapahtumissa, eli näytetään pysyvästi millä tahansa digitaalisella näytöllä. Esimerkiksi CMYK:ä käytetään mekkojen, mainostuotteiden pakkausteollisuudessa. Väritulostimissa on kaksi kasettia, värikasetti ja musta kasetti. Väripatruunassa on syaani, magenta ja keltainen. RGB-värien alkua pidetään mustana, koska valkoinen väri saadaan RGB-värien seoksesta. CMYK-väriyhdyt katsotaan alkavan valkoisella värillä. Koska näiden värien seoksella saadaan mustaa, valkoista ei voida saada.

4.1 Väriin aallonpituus

Valon nopeus on noin 300 miljoonaa metriä (300 000 km) sekuntia. Jokaisella värillä on oma taajuus ja aallonpituus.

Värien taajuudet ja aallonpituudet lasketaan kaavalla:

$$c=f*\lambda.$$

c: Tarkoittaa valonnopeutta ja sen yksikköä metrejä sekunnissa.

f: Tarkoittaa taajuutta ja sen yksikkö on hertsisekunteja.

λ : Viittaa aallonpituuteen ja sen yksikkö on metriä.

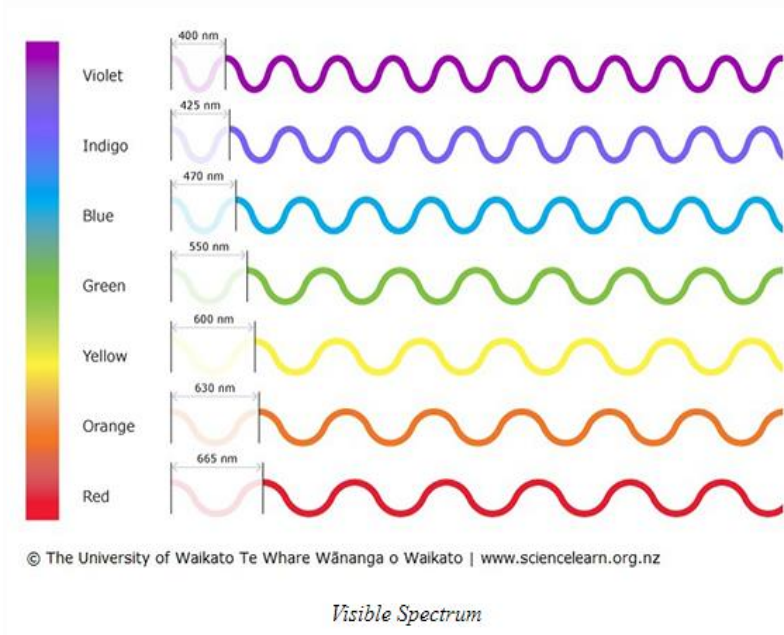
Aallot ovat erittäin tärkeitä ihmiselämässä. Näemme, puhumme ja kuulemme aaltojen avulla. Esimerkiksi voidaan havaita, milloin puhelin on pudonnut maahan sen lähettämien ääniaaltojen ansiosta. Kaikkia aallonpituuksia ei voi nähdä, aallonpituudet, jotka voidaan nähdä, ovat välillä 380nm (nanometri) ja 780nm (nanometri). Kuvassa 12 on väritaulukko väreistä, jotka ihminen havaitsee. (TAOS Colorimetry Tutorial 2006, 2.)

TAULUKKO 2. (mukaillen Table 1 Color vs.Wavelength Range 2006)

Color	Wavelength Range(nm)
Violet	380 ~410
Indigo	410~450
Blue	450~510
Green	510~560
Yellow	560~600
Orange	600~630
Red	630~780

Taulukossa 2 on esitetty väri ja sen aallonpituudet. Näiden värien eri sävyt ovat myös tämän aallonpituusrajan sisällä. On olemassa myös kaksi valoksi luettavaa aallonpituutta, joita ihminen ei voi nähdä. Infrapuna-aalto on lähellä punaisen valon aallonpituutta, mutta siitä ei voi nähdä. Myöskään ultraviolettivaloa ei voi nähdä, mutta se saa ihmisen ihon palamaan. Eri aallonpituuksia käytetään hyvin usein tekniikassa, kuten röntgenkuvissa, televisiokaukosäätimissä ja tiedonsiirtoon.

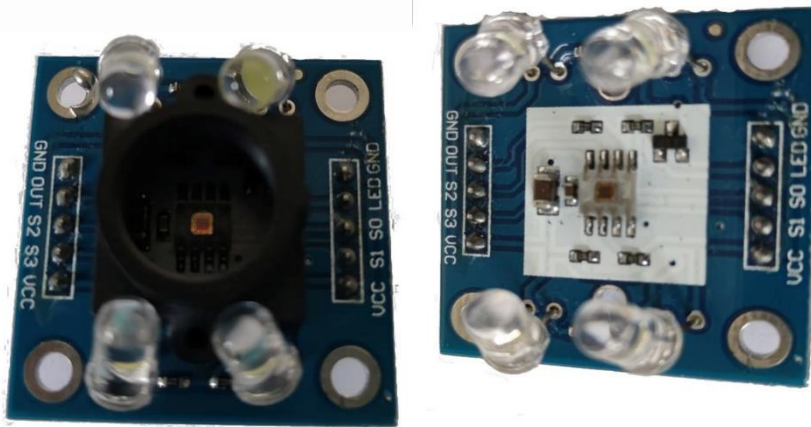
(TAOS Colorimetry Tutorial 2006,2.)



KUVA 12. Valonaalto (mukaillen Colours of light 2022)

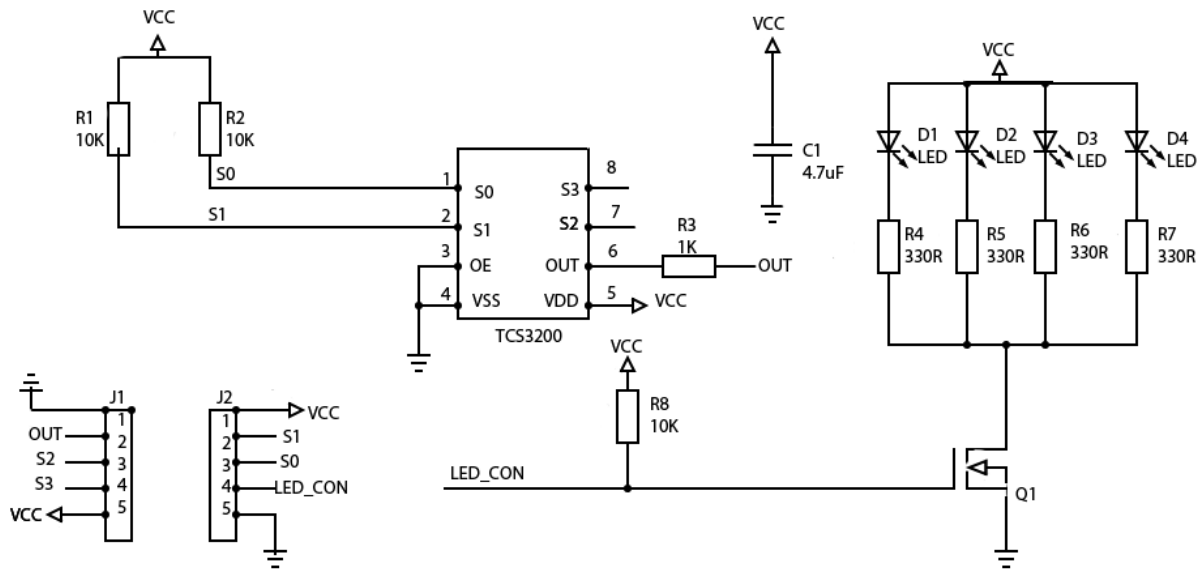
4.2 Väritunnistin moduuli

Tässä osiossa kerrotaan Arduino GY-31 -värimoduulista, jota käytettiin projektissa. On olemassa kaksi yleisimmin käytettyä värianturia, TCS230 ja TCS3200. Näitä myydään eri hintaluokissa. GY-31 moduulissa käytetään Osram Group am5: n myy TCS3200 väritunnistin (KUVA 13. keskellä valkoinen neliö).



KUVA 13. GY-31 Moduuli ja TCS 3200 väritunnistin.

Värianturissa on 8 jalkaa sen kiinnittämiseksi GY-31 moduuliin ja tiedonsiirtoon. Anturissa on liitännät S0, S1, OE, VSS yhdessä osassa ja S3, S2, OUT, VDD toisessa. Anturissa on yhteensä 64 kappaletta valokuvadiodia. Ne on rakennettu 8 riviksi ja 8 sarakkeeksi. Nämä ovat 16 valokuvadiodin punaista värisuodatinta, 16 valokuvadiodin vihreää värisuodatinta, 16 diodisistä värisuodatinta ja 16 valokuvadiodin väritöntä suodatinta. Ne on sijoitettu neliön muotoon $110\mu\text{m} \times 110\mu\text{m}$ alueelle. GY-31 moduulissa on yhteensä 10 nastaa, joista 5 vasemmalla puolella ja 5 oikealla. Ensimmäisessä osassa on GND, OUT, S2, S3, VCC ja toisessa osassa on GND-, LED-, S0-, S1-, VCC-liitännät. Moduulissa käytetään vastusta suojaamaan moduulia suurilta jännitteiltä. Kuviossa 1 näkyy moduulin rakenne ja vastuksen arvot. (TAOS.2006. Colorimetry Tutorial.; TCS3200 Color Sensor Module Schematic.jpg.)



KUVIO 1. GY-31 Moduuli (mukailien TCS3200 Color Sensor Module Schematic.jpg)

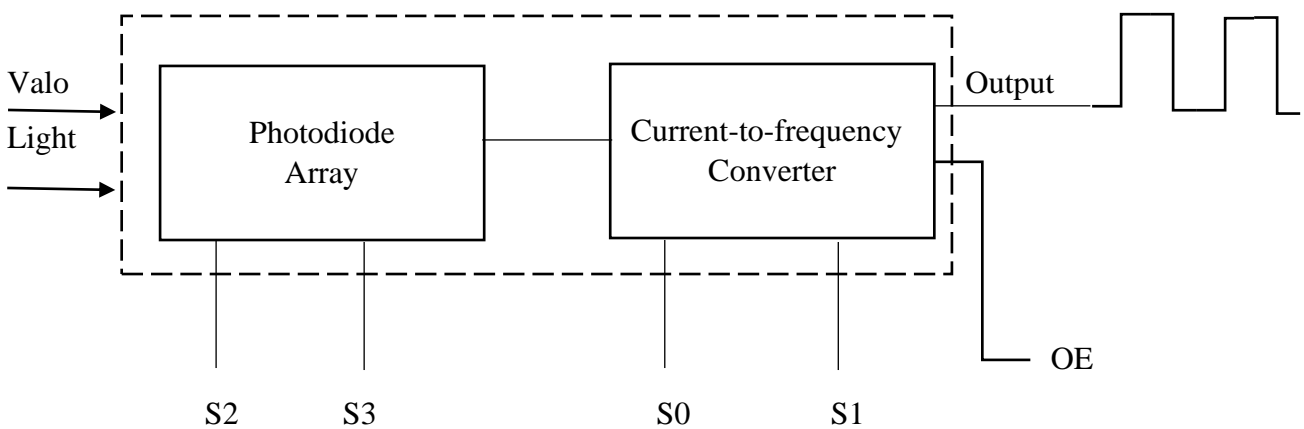
Kuviossa 1. merkinnät tarkoittavat

-VDD(VCC): Jännitevirta.

-GND(Maa): Maataso.

-OUT: Ulos taajuus 1 Kilo-ohmin vastus on kytketty LED-nastaan.

Moduulissa on neljä LED-lamppua. Ledien vastuksia on 330 ohmia per kappale. Anturin S0- ja S1-liittimissä on 10 kilo ohmia vastusta. 1 kilo ohmi vastus on moduulin OUT-nastalla. VCC: n ja GND: n välillä on 4.7 mikro-Faradi-kondensaattoria. Värianturin toimintaperiaate on esitetty kuviossa 2.



KUVIO 2. Funktiodiagrammi (mukailien TCS3200 Block Diagram 2018,2)

Moduuliin tulevat valot saavuttavat ensin valokuvadiodit, väri havaitaan valodiodien värisuodattimissa ja valot muunnetaan taajuuksiksi. Lisäksi moduulissa on digitaaliset tulo- ja digitaaliset lähtöominaisuudet. S2- ja S3-liitäntöjen värityyppi vaihtelee pienjänniteolosuhteiden (LOW) ja suurjänniteolosuhteiden (HIGH) mukaan. Esimerkiksi S2 LOW ja S3 LOW punaiselle värille, S2 HIGH ja S3 HIGH vihreälle värille, S2 LOW ja S3 HIGH siniselle värille, S2 HIGH ja S3 LOW valkoinen värille. S0- ja S1-nastoja käytetään lähtötaajuuksien säätämiseen. Kun S0 on LOW ja S1 on LOW, kun anturi on suljettu (0 %), Kun S0 ON LOW ja S1 ON HIGH, on taajuusasteikko on 2 %, ja kun S0 HIGH ja S1 LOW on asetettu, taajuusasteikko on 20 %. Arvoilla S0 HIGH ja S1 HIGH tehdään täysin tehokas 100%: n taajuusasteikko. Seuraavassa taulukossa on esitetty yhteenveto taajuusasteikosta. (Selectable Options 2018,4)

TAULUKKO 2. Moduulin toiminta (mukailen Selectable Options 2018)

S0	S1	Output Frequency Ulos taajuus	S2	S2	Photodiode Type (Valodiodin tyyppi)
Low	Low	Power down	Low	Low	Red (Punainen)
Low	High	2 %	Low	High	Blue (Sininen)
High	Low	20 %	High	Low	Clear (Kirkas)
High	High	100 %	High	High	Green (Vihreä)

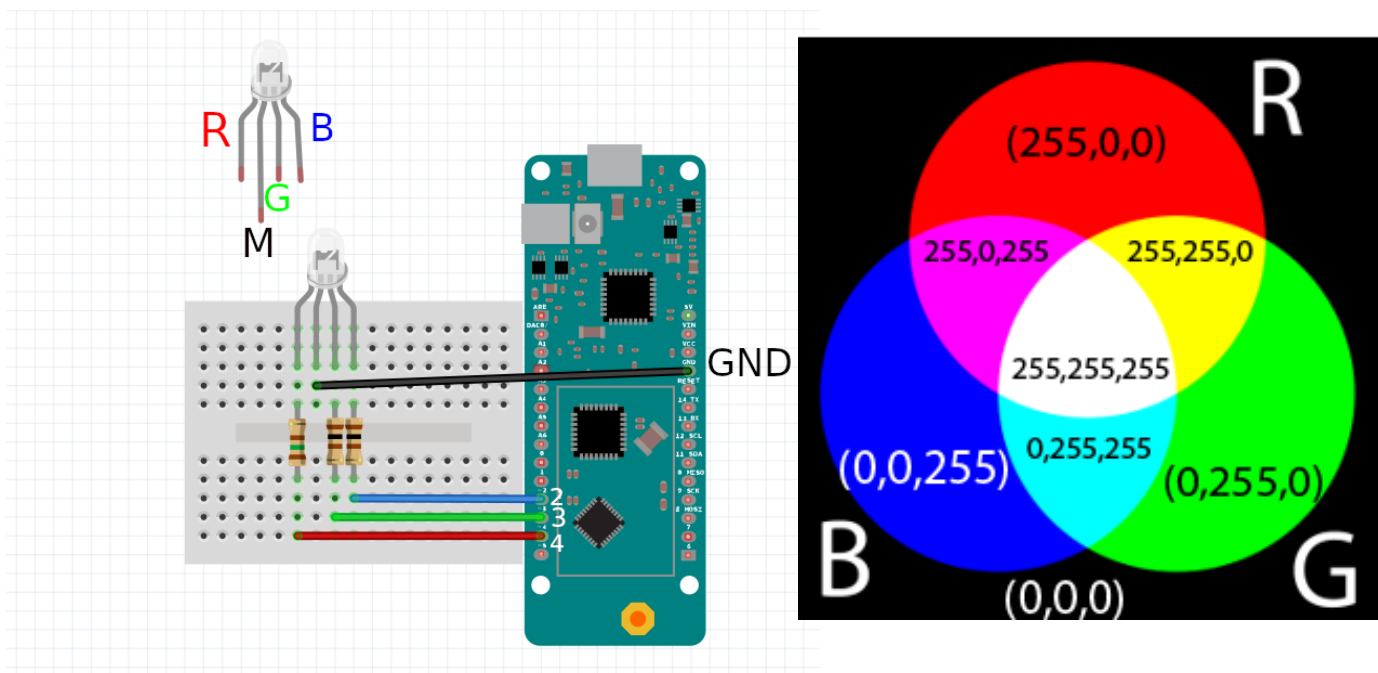
Arduino-laitteiden suositeltu taajuusasteikko on 20 % (S0 HIGH, S1 LOW). Värianturi maksivirta on 5 voltia, jota tarvitaan optimaaliseen toimintaan, mutta laite voi toimia välillä 2,7 ja 5,5 voltia. Sääolosuhteet vaikuttavat laitteen ihanteelliseen toimintaan. 25 C on ihanteellinen, mutta alaraja on -40 C ja yläraja on 70 C. (Recommended Operating Conditions 2018,6)

5 KOEJÄRJESTELYT

Koejärjestelyissä LED, Arduino-värianturi ja Sigfox-moduuli testataan erikseen. Ensiksi LED-kytkentä ja Arduino-laitteen kytkentäkaavio piirrettiin Fritzing-ohjelmalla ja koodattiin toimimaan Arduino-IDE:ssä. Sitten Arduino-laite ja värianturi kytketään toimintaan, aluksi Fritzing-ohjelmassa, ja koodattiin toimimaan Arduino-IDE:ssä. Kun LED ja värianturi toimivat erikseen, yhdistetään ne kokonaisuudeksi, Fritzing-ohjelman avulla piirrettiin kokonainen kytkentäkaavio ja lopullinen koodi toteutetaan.

5.1 RGB-led testi

RGB-ledillä on neljä metallista jalkaa. Ensimmäinen metallien jalka on Red (punainen), toinen metallinen jalka Ground (M)(maataso), kolmas metallinen jalka Green (punainen) ja neljäs metallinen jalka on Blue (sininen). RGB-lampun kytkentäkaavio tehtiin Fritzing-ohjelmalla. Se on simulaattoriohjelma, jolla suunnitellaan ja koodataan Arduino-projekteja. Fritzing on ilmainen avoimen lähdekoodin ohjelma. Kuviossa 3 on esitetty RGB-led koekytkentä.



KUVIO 3. Vasemmalla RGB-led koe piiri, oikealla RGB värit.

Kuviossa 3 esitetään, kuinka RGB-LED-valojen jalat on kytketty Arduino-kortin 2,3,4- ja GND-nastoihin. LED-jalkojen ja Arduino-kortin välillä käytetään vastuksia, koska korkean virta voi rikkoa tai häiritä LED-valoja. Punaiseen LED-jalkaan on kiinnitetty 150 ohmin vastus ja 100 ohmin vastus on kiinnitetty vihreään ja siniseen LED-jalkaan. Kun punainen LED on kytketty vastukseen, sen jälkeen vastuksen jalka kytketään Arduino-kortin nastaan neljä. Vihreä LED kytketty vastukseen ja Arduino-kortin välille, vastuksen jalka kytketään Arduino-kortin nastaan kaksi. LED-valon maataso-jalka on kytketty suoran Arduino-kortin GND nastaan. Alla on RGB-LED:n testin osittainen koodi, tällä esimerkillä voidaan sytyttää punainen LED. RGB-LED:n testin koko koodi on esitetty LIITE1/1 ja LIITE1/2.

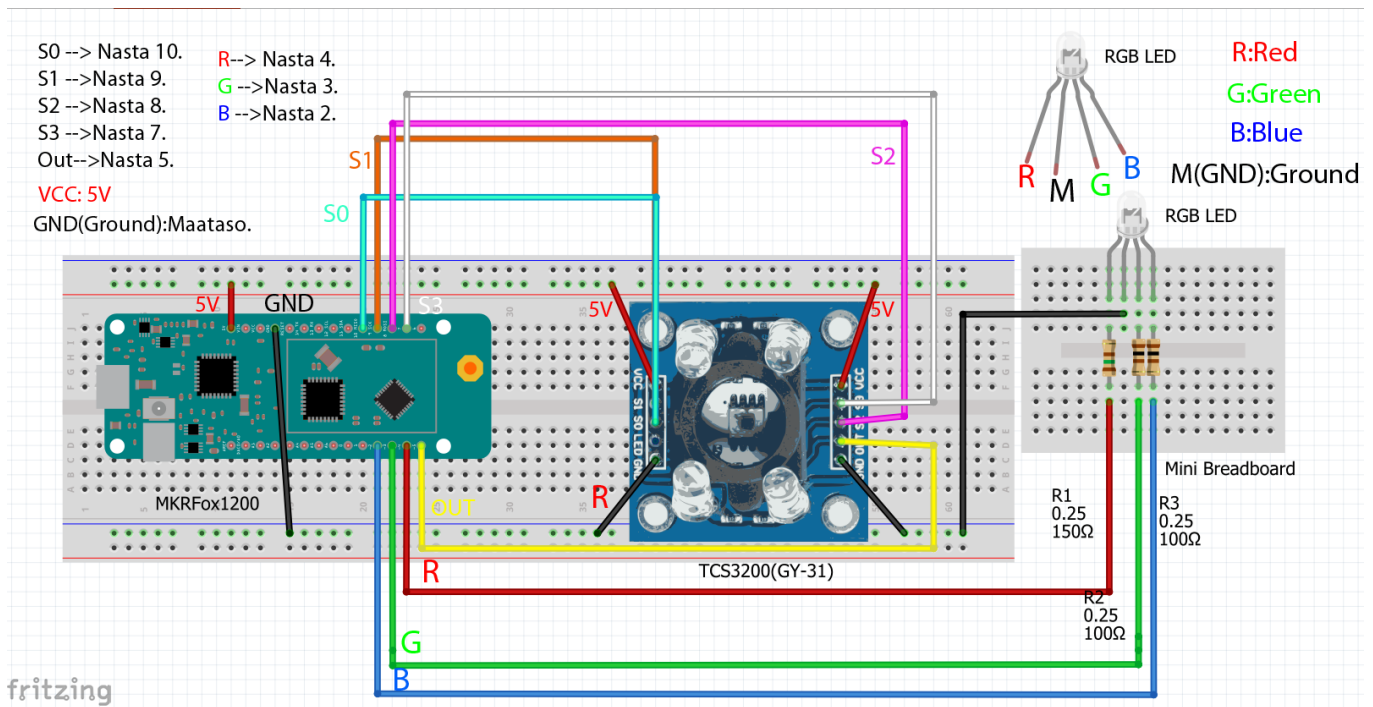
```
#define rled 4 //Punainen ledi kytketään nastaan neljä.
void setup() { //Tässä tehdään nastan tulo- tai lähtö-
asetuksia.
pinMode(rled, OUTPUT); //Määritellään punaisen nastan lähtö.
Serial.begin(9600); //Sarjamonitoru avataan ja määritellään
nopeus 9600 bps(bitti per sekuntti)
}

void loop() { //Silmukka funktio(tehdään silmukka).
Serial.println("Red "); // Sarjamonitorissa tulostetaan Red
teksti.
analogWrite(rled,255); //Punainen ledi sytytetään.
delay(1000); // Tuhannen millisekunnin viive.
}
```

Arduino-kortin nastaa neljä kutsutaan rlediksi koodauksen helpottamiseksi. Void setup -funktion sarjaportti avataan sarjamonitorille ja tiedonsiirtonopeudeksi on asetettu 9600 bps(bitti per sekunti). rled-nastalle määritellään lähtö. Void loop -silmukassa toiminnossa LED:n arvo asetetaan ja sarjamonitoriin tulostetaan "Red"-teksti. Tämän jälkeen odotetaan 1000 millisekuntia. RGB-testaukseen tarvittavat koodit ovat LIITE1/1 ja LIITE1/2.

5.2 TCS3200(GY-31) värianturi ja RGB led testi

Tässä osiossa RGB-led ja Arduino-värianturin kytketään samaan kytkentäkaavioon. Kun kytkentäkaavio on piirretty Fritzing-ohjelmalla, aloitetaan koodaaminen Arduino-IDE ohjelmalla.



KUVIO 4. RGB-led ja väritunnustuin kytkentäkaavio. (Batman 2022)

S0 kytketään nastaan kymmenen.

S1 kytketään nastaan yhdeksän.

S2 kytketään nastaan kahdeksan.

S3 kytketään nastaan seitsemän.

OUT kytketään nastaan viisi.

Punainen ledi kytketään nastaan neljä.

Vihreä ledi kytketään nastaan kolme.

Sininen ledi kytketään nastaan kaksi.

Koodissa määritellään aluksi arvot eri muuttujille

```
//Alussa määritellään punaisen, -vihreän, -sinisen värien arvoja nollaksi.
```



```
int red, green, blue = 0;
```

int r_min_ta=970; // Väritynnistin ensimmäisen kalibrointi funktio (do_calibration()) jälkeen, määritetään punaisen värin minimi arvo. int r_max_ta=57; // Väritynnistin ensimmäisen kalibrointi (do_calibration()) jälkeen, määritetään punaisen värin maksimi arvo.

void setup() -funktioissa tehdään nastan tulo- tai lähtöasetukset sekä avataan sarjamonitori ja asetetaan tiedonsiirto nopeus.

```
void setup
{
//Määritellään S0:n nastan lähtö.
    pinMode(s0, OUTPUT);
    // Sarjamonitori avataan ja määritellään nopeus 9600 bps
    Serial.begin(9600);
}
```

Silmukka-funktio (void loop()) sisälle on koodattu kolme funktiota. Ne ovat do_calibration(), did_calibration(), ja colordetect() .

do_calibration(): Kalibrointi tehdään RGB-värin maksimi- ja minimiarvon määrittämistä varten valkoisella ja mustalla paperilla.

did_calibration(): Toinen kalibrointifunktio käytetään värin tunnistamista varten, tällöin käytetään RGB-värisiä papereita.

colordetect(): Väritynnistuksen tekevä funktio (tunnistaa värin ja sytyttää RGB-ledin).

Esimerkki koodi: do_calibration() funktio, jonka osa.

```
void do_calibration() { //Kalibrointi tehdään funktion värin maksimi- ja minimiarvon varten.
    red = pulseIn(out, LOW); //Saadaan punainen värinimpulssi väritynnistimelta.
}
```

Esimerkki koodi: did_calibration() funktio, jonka osa.

```
void did_calibration() { //Toinen kalibrointi funktion värin arvon käännoystä varten    red = pulseIn(out, LOW);
    //Saadaan punaisen värin impulssi väritynnistimelta.
    //red = map(red, 970, 57, 0, 255);
    red = map(red, r_min_ta, r_max_ta, 0, 255); //Punaisen värin maksimi- ja minimiarvon
```

```
}

```

Esimerkki koodi: colordetect() funktion, jonka osa.

```
void colordetect() { //Värin pulssin arvolla määritellään väri ja ledin väri.
    // Funktion ehdot: Jos if funktion ehto on tosi, hakasulun sisällä koodi toimitetaan.
    if (red>=237 and red<=257 and green>=154 and green<=174 and blue>=167 and blue<=187) {
        Serial.println("Detected Red "); // Sarjamonitorissa tulostetaan Detected Red teksti.
        analogWrite(rled,255); //Punainen ledi sytytetään.
    }
    Void loop(){ //Ensimmäinen kalibrointisilmukka
do_calibration();
//did_calibration();
//colordetect();
}

    Void loop(){ //Valmis kalibrointisilmukka
//do_calibration();
did_calibration();
//colordetect();
}

Void loop(){ //Valmis kalibrointisilmukka ja väritunnistin funktio

//do_calibration();
did_calibration();
colordetect();
}

```

TCS3200(GY-31) värianturi ja RGB-LED testaukseen tarvittavat koodit ovat esitettyinä LIITE 2.

6 JOHTOPÄÄTÖKSET

Projektin tavoitteena oli selittää yksinkertaisella tavalla, miten Arduino-IDE-ohjelmaa käytetään ja opastaa lukijalle, kuinka voi tehdä Arduino-projekteja. Apuna käytettiin RGB-LEDiä värin näyttämiseen sekä väritunnistamiseen. Lisäksi hyödynnettiin Sigfox-verkkoyhteyttä viestin lähettämiseen palvelimelle. Suurin osa tavoitteista saavutettiin, mutta Arduino Sigfox -moduulia ei saatu kunnolla toimimaan, ensin kirjasto-ongelmien vuoksi ja myöhemmin lisensseihin liittyvien ongelmien takia. Sigfox-viestin lähetys toimi kuitenkin kooditasolla, eikä työssä haluttu keskittyä tähän liiaksi, sillä toteutus haluttiin pitää suoraviivaisena ja selkeänä.

Toteutuksessa tehtiin Arduinon-värianturilla testejä käyttäen RGB-LED-valoa, työn koodit kirjoitettiin yksinkertaiseksi ja ymmärrettäväksi- RGB-LED-valoa oli helppo säätää ja työssä saatiin päävärit ja välivärit toimimaan. Kytkeäntöjen suunnitteluun käytettiin Fritzing-ohjelmaa, tästä oli hyötyä projektin kytkeäntäkaavioiden piirtämisessä. Fritzing-ohjelmalla voidaan suunnitella useimmille Arduino-laitteille, mutta MKRFOX1200-perheen laitteille ei ollut tukea, vaan tämä piti asentaa erikseen.

Ensimmäisessä liitäntäkaaviossa käytettiin VCC-nastaa sähkövirralle, VCC antoi virtaa välillä 2,7–5 V, mutta virta oli alle 5 V: n. Tämä virta ei riittänyt värianturille. Seuraavaksi käytettiin MKR-FOX1200 Arduino -kortin 5 V-nastaa värianturin virraksi, mikä toimi. Arduino-IDE versio 1.8.16 aiheutti myös hieman ongelmia sarjamonitoriosassa sen jälkeen, tämä ilmeni, kun Sigfox-moduulin kirjasto oli lisätty projektiin. Päivitetyssä Arduino IDE 2.0:ssa sarjamonitori toimi sujuvasti.

Värianturi vaatii toimiakseen kalibroinnin, kalibroinnin jälkeen anturi tuottaa tasaisia tuloksia, mutta on herkkä muutoksille ja etäisyydelle. Anturi vaatiikin toimiakseen oikean ympäristön.

LÄHTEET

Amazon.com. 2022. Saatavissa: https://www.amazon.com/EL-KIT-001-Project-Complete-Starter-Tutorial/dp/B01CZTLHGE/ref=sr_1_4?crid=19I6T01YIYN85&keywords=elegoo+uno+project+super+starter+kit&qid=1650185517&prefix=elegoo+uno%2Caps%2C264&sr=8-4. Viitattu 17.4.2022.

Arduino Integrated Development Environment (IDE) v1. 2022.Saatavissa: <https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics>. Viitattu 16.4.2022.

Arduino. 2018.What is Arduino? . Saatavissa: <https://www.arduino.cc/en/Guide/Introduction>. Viitattu 4.4.2022.

Arduino. 2020. Language Reference. Saatavissa: <https://www.arduino.cc/reference/en/>. Viitattu 1.5.2022.

Arduino. 2021.Our Mission & Vision. Saatavissa: <https://www.arduino.cc/en/about#our-mission--vision>. Viitattu 5.4.2022.

Arduino. 2022. Hardware. Saatavissa: <https://www.arduino.cc/en/hardware>. Viitattu 17.4.2022.

Arduino. 2022.MKR FOX 1200. Saatavissa: <https://docs.arduino.cc/hardware/mkr-fox-1200>. Viitattu 13.4.2022.

Arduino 2022.Products. Entry Level. Saatavissa: <https://www.arduino.cc/en/Main/Products>. Viitattu 6.4.2022.

Arduino Starter Kit Multi-language. 2021. Saatavissa: <https://store.arduino.cc/products/arduino-starter-kit-multi-language>. Viitattu 17.4.2022.

Colour mixing. 2022. Saatavissa: <https://www.sciencelearn.org.nz/images/44-colour-mixing>. Viitattu 25.4.2024

Colours of light .2022.Saatavissa: <https://www.sciencelearn.org.nz/resources/47-colours-of-light>. Viitattu 25.4.2022.

ELEGOO UNO R3 Project Most Complete Starter Kit.2022. Saatavissa: <https://www.elegoo.com/collections/uno-r3-starter-kits/products/elegoo-uno-most-complete-starter-kit>. Viitattu 17.2.2022.

IoT use cases for agriculture. Saatavissa: https://connectedinventions.com/wp-content/uploads/2021/08/CFSigfox_UseCase_for_Agriculture.pdf. Viitattu 19.4.2022.

Isohanni J.2017.Sigfox – maailmanlaajuinen IoT-verkkoliittymä. Saatavissa <https://centriabulletin.fi/sigfox-maailmanlaajuinen-iot-verkkoliittyma/>. Viitattu 19.4.2022.

Newton's Rainbow. Saatavissa: <https://library.si.edu/exhibition/color-in-a-new-light/science>. Viitattu 23.4.2022.

Our Story. Saatavissa <https://www.sigfox.com/en/sigfox-story>. Viitattu 19.4.2022.

Overview of the Arduino IDE 1.2022. Saatavissa: <https://docs.arduino.cc/software/ide-v1/tutorials/Environment>. Viitattu 7.4.2022.

pixabay Lightshow Matryx. Saatavissa: <https://pixabay.com/illustrations/lightshow-rainbow-lighting-light-5968796/>. Viitattu 25.4.2022.

Recommended Operating Conditions 2018. Saatavissa: https://ams.com/documents/20143/36005/TCS3200_DS000107_3-00.pdf/c53d702f-63aa-eda1-745f-d513aa4f535f . Viitattu 23.4.2022.

RGB OpenClipart-Vectors .2022. Saatavissa: <https://pixabay.com/vectors/intersection-mix-colors-rgb-red-154782/>. Viitattu 25.4.2022.

RGB vs CMYK: What's the difference?.2019. Saatavissa: <https://99designs.com/blog/tips/correct-file-formats-rgb-and-cmyk/>. Viitattu 23.4.2022.

Selectable Options 2018. Saatavissa: https://ams.com/documents/20143/36005/TCS3200_DS000107_3-00.pdf/c53d702f-63aa-eda1-745f-d513aa4f535f. Viitattu 22.4.2022.

Sigfox Tech Overview and Resources. Saatavissa: <https://www.connectedfinland.fi/aloita-tasta/kehittajat/>. Viitattu 19.4.2022.

Table 1 Color vs.Wavelength Range .2006 Saatavissa: https://ams.com/documents/20143/36005/LightSensors_AN000519_1-00.pdf/d2a0670d-7557-ed94-cc0e-225cfa8cb031. Viitattu 24.4.2022

TAOS. 2006. Colorimetry Tutorial. Saatavissa: https://ams.com/documents/20143/36005/LightSensors_AN000519_1-00.pdf/d2a0670d-7557-ed94-cc0e-225cfa8cb031. Viitattu 24.04.2022

TCS3200 Color Sensor Module Schematic.jpg. Saatavissa:<https://www.openimpulse.com/blog/wp-content/uploads/wpsc/downloadables/TCS3200-Color-Sensor-Module-Schematic.jpg>. Viitattu 22.4.2022

TCS3200 Block Diagram .2018. Saatavissa: <https://ams.com/en/tcs3200>. Viitattu 22.4.2022.


```

#define s0 10          // S0 kytketään nastaan kymmenen.
#define s1 9          // S1 kytketään nastaan yhdeksän.
#define s2 8          // S2 kytketään nastaan kahdeksan.
#define s3 7          // S3 kytketään nastaan seitsemän.
#define out 5         // OUT kytketään nastaan viisi.
#define rled 4        // Punainen ledi kytketään nastaan neljä.
#define gled 3        // Vihreä ledi kytketään nastaan kolme.
#define bled 2        // Sininen ledi kytketään nastaan kaksi.

int red, green, blue = 0; //Alussa määritellään punaisen, -vihreän, -sinisen värien arvoja nolllaksi.
int r_min_ta=970; // Väritunnistin ensimmäisen kalibrointi jälkeen, määritetään punaisen värin minimi
//arvo.
int r_max_ta=57; // Väritunnistin ensimmäisen kalibrointi jälkeen, määritetään punaisen värin maksimi
//arvo.
int g_min_ta=1000; // Väritunnistin ensimmäisen kalibrointi jälkeen, määritetään vihreän värin minimi
//arvo.
int g_max_ta=61; // Väritunnistin ensimmäisen kalibrointi jälkeen, määritetään vihreän värin maksimi
//arvo.
int b_min_ta=810; // Väritunnistin ensimmäisen kalibrointi jälkeen, määritetään sinisen värin minimi
//arvo.
int b_max_ta=49; // Väritunnistin ensimmäisen kalibrointi jälkeen, määritetään sinisen värin maksimi
//arvo.

void setup() {          //Tässä tehdään nastan tulo- tai lähtöasetuksia.
  pinMode(s0, OUTPUT); //Määritellään S0:n nastan lähtö.
  pinMode(s1, OUTPUT); //Määritellään S1:n nastan lähtö.
  pinMode(s2, OUTPUT); //Määritellään S2:n nastan lähtö.
  pinMode(s3, OUTPUT); //Määritellään S3:n n nastan lähtö.
  pinMode(out, INPUT); //Määritellään OUT nastan tulo.

```



```

pinMode(rlcd, OUTPUT); //Määritellään punaisen nastan lähtö.
pinMode(gled, OUTPUT); //Määritellään vihreän nastan lähtö.
pinMode(bled, OUTPUT); //Määritellään sinisen nastan lähtö.

digitalWrite(s0, HIGH); //Määritellään S0:n nastan tosi(HIGH).
digitalWrite(s1, LOW); //Määritellään S1:n nastan epätosi(LOW).
Serial.begin(9600); //Sarjamonitori avataan ja määritellään nopeus 9600 bps(bitti
//per sekunti)
}

void loop() { //Silmukka funktio(tehdään silmukka).
//do_calibration(); //Kalibrointi tehdään funktio RGB värin maksimi ja minimi
//arvon varten.
did_calibration(); //Kalibrointi tehdään funktio värejä varten.
colordetect(); //Väritunnistin funktio(tunnistaa värejä ja sytyttää RGB
//ledin).

}

void do_calibration() { //Kalibrointi tehdään funktioni värin maksimi ja minimi arvon varten.

// Tässä tehdään väritunnistimen punaisen värin asetuksia.
digitalWrite(s2, LOW); //Määritellään S2:n nastan epätosi(LOW).
digitalWrite(s3, LOW); //Määritellään S3:n nastan epätosi(LOW).
red = pulseIn(out, LOW); //Saadaan punainen värin impulssi väritunnistimelta.
Serial.print("Red: "); // Sarjamonitorissa tulostetaan Red teksti.
Serial.print(red); // Sarjamonitorissa tulostetaan punaisen värin impulssi arvo.
Serial.print("\t"); // Sarjamonitorissa tulostetaan tabulaattori.
delay(100); // Sadan millisekunnin viive.

// Tässä tehdään väritunnistimen vihreän värin asetuksia.
digitalWrite(s2, HIGH); //Määritellään S2:n nastan tosi(HIGH).

```

```

digitalWrite(s3, HIGH); //Määritellään S3:n nastan tosi(HIGH).
green = pulseIn(out, LOW); //Saadaan vihreän värin impulssi väritunnistimelta.
Serial.print("Green: "); // Sarjamonitorissa tulostetaan Green teksti.
Serial.print(green); // Sarjamonitorissa tulostetaan vihreän värin impulssi arvo.
Serial.print("\t"); // Sarjamonitorissa tulostetaan tabulaattori.
delay(100); // Sadan millisekunnin viive.

// Tässä tehdään väritunnistin sinisen värin asetuksia.
digitalWrite(s2, LOW); //Määritellään S2:n nastan epätosi(LOW).
digitalWrite(s3, HIGH); //Määritellään S3:n nastan tosi(HIGH).
blue = pulseIn(out, LOW); //Saadaan sinisen värin impulssi väritunnistimelta.
Serial.print("Blue: "); // Sarjamonitorissa tulostetaan Blue teksti.
Serial.print(blue); // Sarjamonitorissa tulostetaan sinisen värin impulssi arvo.
Serial.println("\t"); // Sarjamonitorissa tulostetaan tabulaattori.
delay(100); // Sadan millisekunnin viive.
}
void did_calibration() { //Toinen kalibrointi funktioni, värin arvon käännöstä varten.

// Tässä tehdään väritunnistin punaisen värin asetuksia.
digitalWrite(s2, LOW); //Määritellään S2:n nastan epätosi(LOW).
digitalWrite(s3, LOW); //Määritellään S3:n nastan epätosi(LOW).
red = pulseIn(out, LOW); //Saadaan punaisen värin impulssi väritunnistimelta.
//red = map(red, 970, 57, 0, 255);
red = map(red, r_min_ta,r_max_ta, 0, 255); //Punaisen värin maksimi ja minimi arvo
// käännetään map funktioni avulla.
Serial.print("Red color: "); // Sarjamonitorissa tulostetaan Red teksti.
Serial.print(red); // Sarjamonitorissa tulostetaan punaisen värin impulssi arvo.
Serial.print("\t"); // Sarjamonitorissa tulostetaan tabulaattori.
delay(100); // Sadan millisekunnin viive.

// Tässä tehdään väritunnistin vihreän värin asetuksia.
digitalWrite(s2, HIGH); //Määritellään S2:n nastan tosi(HIGH).

```

```

digitalWrite(s3, HIGH);          //Määritellään S3:n nastan tosi(HIGH).
  green = pulseIn(out, LOW);     //Saadaan vihreän värin impulssi väritunnistimelta.
  //green = map(green, 1000, 61, 0, 255);
  green = map(green, g_min_ta,g_max_ta, 0, 255); //Vihreän värin maksimi ja minimi arvo
// käännetään map funktion avulla.
  Serial.print("Green color: "); // Sarjamonitorissa tulostetaan Green teksti.
  Serial.print(green);          // Sarjamonitorissa tulostetaan vihreän värin impulssi arvo.
  Serial.print("\t");           // Sarjamonitorissa tulostetaan tabulaattori.
  delay(100);                   // Sadan millisekunnin viive.

                                // Tässä tehdään väritunnistin sinisen värin asetuksia.
digitalWrite(s2, LOW);          //Määritellään S2:n nastan epätosi(LOW).
digitalWrite(s3, HIGH);         //Määritellään S3:n nastan tosi(HIGH).
  blue = pulseIn(out, LOW);     //Saadaan sinisen värin impulssi väritunnistimelta.
  //blue = map(blue, 810,49, 0, 255);
  blue = map(blue, b_min_ta,b_max_ta, 0, 255);
  Serial.print("Blue color: "); // Sarjamonitorissa tulostetaan Blue teksti.
  Serial.print(blue);          // Sarjamonitorissa tulostetaan sinisen värin impulssi arvo.
  Serial.println("\t");        // Sarjamonitorissa tulostetaan tabulattori.
  delay(100);
}
void colordetect() {           //Värin pulssin arvolla märitellään väri ja ledin väri.

  // Funktion ehdot: Jos if funktion ehto on tosi, hakasulun sisällä koodi toimitetaan.
  if (red>=237 and red<=257 and green>=154 and green<=174 and blue>=167 and blue<=187) {
    Serial.println("Detected Red "); // Sarjamonitorissa tulostetaan Detected Red teksti.
    analogWrite(rlcd,255);          //Punainen ledi sytytetään.
    analogWrite(gled,0);            // Vihreä ledi sammutetaan.
    analogWrite(bled,0);            // Sininen ledi sammutetaan.
  }
  else if (red>=202 and red<=222 and green>=227 and green<=247 and blue>=212 and blue<=232 )
  {
    Serial.println("Detected Green "); // Sarjamonitorissa tulostetaan Detected Green teksti.

```

```

analogWrite(rled,0);           //Punainen ledi sammutetaan.
analogWrite(gled,255);        // Vihreä valo sytytetään
analogWrite(bled,0);          // Sininen valo sammutetaan.
}
else if (red>=90 and red<=112 and green>=158 and green<=178 and blue>=208 and blue<=228 )
{
  Serial.println("Detected Blue "); // Sarjamonitorissa tulostetaan Detected Blue teksti.
  analogWrite(rled,0);           //Punainen ledi sammutetaan.
  analogWrite(gled,0);           // Vihreä ledi sammutetaan.
  analogWrite(bled,255);         // Sininen ledi sytytetään.
}
else if (red>=245 and red<=265 and green>=240 and green<=260 and blue>=222 and blue<242 )
{
  Serial.println("Detected Yellow "); // Sarjamonitorissa tulostetaan Detected Yellow teksti.
  analogWrite(rled,255);         //Punainen ledi sytytetään.
  analogWrite(gled,255);         // Vihreä ledi sytytetään.
  analogWrite(bled,0);           // Sininen ledi sammutetaan.
}

else if (red>=251 and red<=271 and green>=219 and green<=239 and blue>=235 and blue<=255 )
{
  Serial.println("Detected Pink "); // Sarjamonitorissa tulostetaan Detected Pink teksti.
  analogWrite(rled,255);         //Punainen ledi sytytetään.
  analogWrite(gled,0);           // Vihreä valo sammutetaan.
  analogWrite(bled,255);         // Sininen valo sytytetään.
}

else if (red>=211 and red<=231 and green>=236 and green<=256 and blue>=241 and blue<=261 )
{
  Serial.println("Detected Aqua Blue "); // Sarjamonitorissa tulostetaan Detected Aqua Blue teksti.

  analogWrite(rled,0);           //Punainen ledi sammutetaan.
  analogWrite(gled,255);         // Vihreä valo sytytetään.

```

```
analogWrite(bled,255);          // Sininen valo sytytetään.
}

else if (red>=244 and red<=264 and green>=243 and green<=263 and blue>=244 and blue<=264 )
{
  Serial.println("Detected White "); // Sarjamonitorissa tulostetaan Detected White teksti.
  analogWrite(rled,255);           // Punainen valo sytytetään
  analogWrite(gled,255);           // Vihreä valo sytytetään.
  analogWrite(bled,255);           // Sininen valo sytytetään.
}

else if (red>=-18 and red<=2 and green>=-19 and green<=1 and blue>=-20 and blue<=0 ) {
  Serial.println("Detected Black"); // Sarjamonitorissa tulostetaan Detected Black teksti.
  analogWrite(rled,0);             //Punainen ledi sammutetaan.
  analogWrite(gled,0);             //Vihreä ledi sammutetaan.
  analogWrite(bled,0);             // Sininen valo sammutetaan.
}

else{
  Serial.println("Color not detechted"); // Sarjamonitorissa tulostetaan Color not Detected teksti.
  analogWrite(rled,0);             //Punainen ledi sammutetaan.
  analogWrite(gled,0);             //Vihreä ledi sammutetaan.
  analogWrite(bled,0);             // Sininen valo sammutetaan.
}

}
```