

INF1510

Cyberplant

TEKNISK
DOKUMENTASJON

Luna Wei Shen (lunas) Herman Loennechen (hermanlo)
Victor Risopatron (victori)

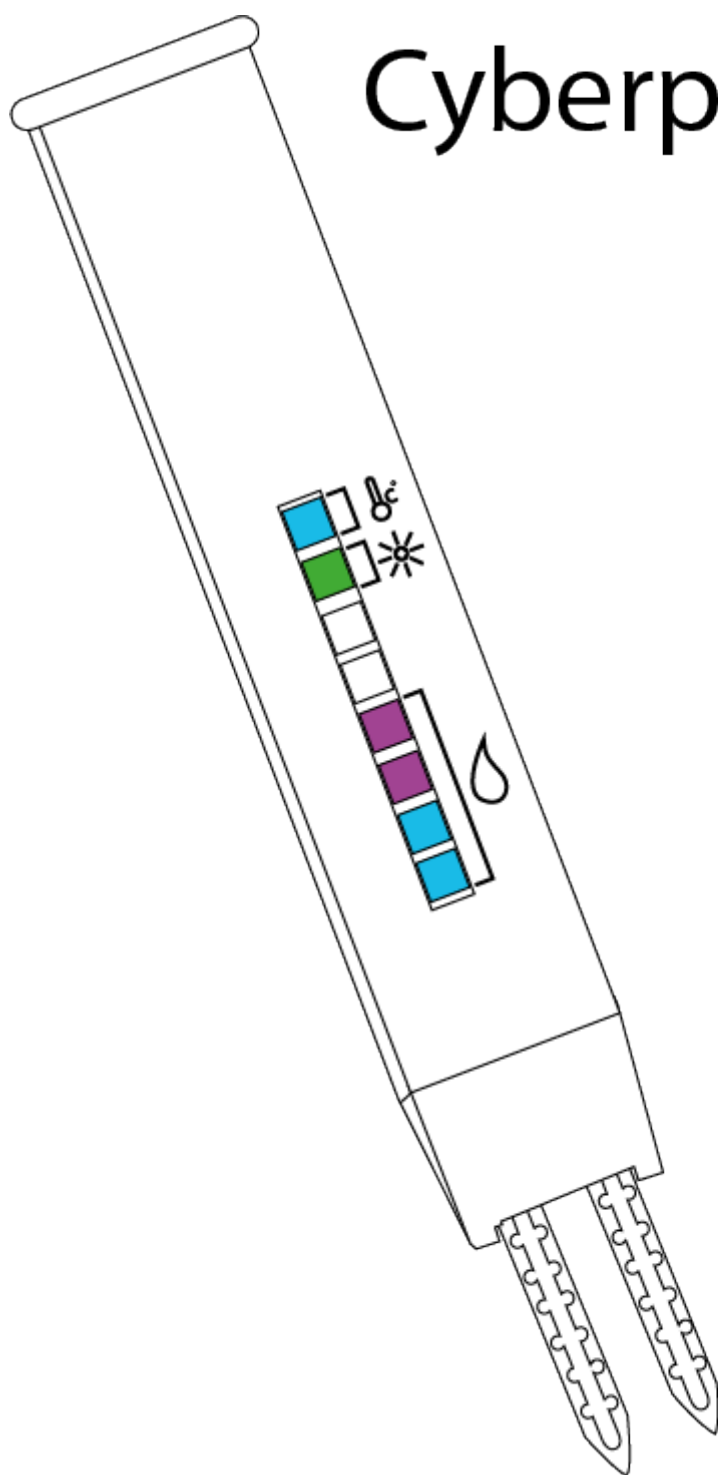


The Faculty of Mathematics and Natural Sciences
University of Oslo
Norway

Innhold

1	Innledning	4
2	Video	4
3	Teknisk dokumentasjon av Cyberplant	4
3.1	Komponenter	5
3.2	Funksjoner	5
4	Kode	5
4.1	Arduinokode	5
5	Skjemaer og Datasheets	11
5.1	Arduino Nano	11
5.2	ESP8266	12
5.3	TSL2561	12
5.4	WS2812b	13
5.5	YL-69 og YL-38	14
5.6	TMP36(GZ)	14

Cyberplant



1 Innledning

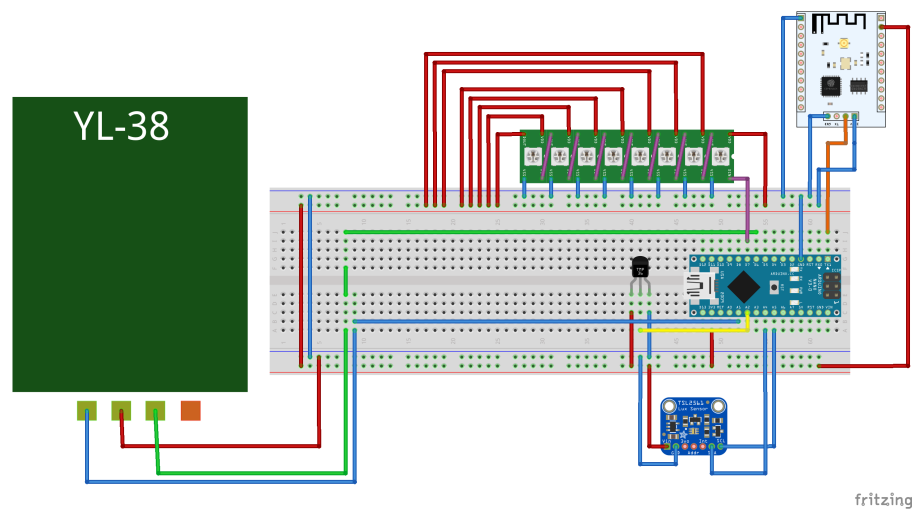
Målet for prosjektet er å lage et produkt til vår målgruppe med problemstillingen "Hvordan kan vi hjelpe hobbygartnere med overvåkning av plantenes helse og innsamling av plantedata?". Vår målgruppe ønsker en artifakt som kan hjelpe han med hage arbeidet og vårt mål er å lage en teknisk løsning som kan hjelpe han. Prototypen vår vil hjelpe vår målgruppe å se hvordan planten har det med tanke på vann, temperatur og lys til enhver tid og bidra med å gi han informasjon til hva som må gjøres for å gi planten bedre leveforhold.

2 Video

Videoen viser utviklingen av Cyberplant fra tidlig konsept til den endelige prototypen, 1.gangsinteraksjon av målgruppe og demonstrasjon av funksjonene.

Videolink: <https://www.youtube.com/watch?v=sUky9X-d9wo&feature=youtu.be>

3 Teknisk dokumentasjon av Cyberplant



Figur 1: Oppkobling av krets

3.1 Komponenter

Arduino Nano

ESP8266/ESP12e/NodeMCU dev board

Lux-sensor TSL2561

WS2812b LED-strip 8stk

Soil Moisture Sensor YL-69

Soil Moisture Sensor kontroller YL-38

Temperature sensor TMP36 (36GZ)

6V driftspenning via 2 stk 3V CR2 lithium batterier.

3.2 Funksjoner

Cyberplant har følgende funksjoner:

Lyssensor fra 0-17000 Lux med varsling via LED-lys.

Temperatursensor fra -55°C til 150°C med varsling via LED-lys.

Jordfuktighet/humidity med varsling via LED-lys.

Støtte for seriell dataoverføring via kabel eller WiFi.

6-10 Volts driftspenning.

4 Kode

4.1 Arduinokode

Listing 1: Some Code

```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_TSL2561_U.h>
#include <Adafruit_NeoPixel.h>

byte humidity_sensor_pin = A1;
byte humidity_sensor_vcc = 6;
int tempPin = A2;
int PIN = 7;
int totalLEDs = 8;
int delayval = 250;
int val = 0; // testvalue hum sensor to wifi module

Adafruit_NeoPixel leds = Adafruit_NeoPixel(totalLEDs, PIN,
    ↪ NEO_GRB + NEO_KHZ800);

Adafruit_TSL2561_Unified tsl = Adafruit_TSL2561_Unified(
    ↪ TSL2561_ADDR_FLOAT, 12345);

void setup() {

    // Setup Serial
    while (!Serial);
```

```

delay(1000);
Serial.begin(9600);

// Initialise LEDs
leds.begin();
leds.setBrightness(50); // sets leds to strength value
    ↪ between 0-255
leds.show(); // Initialize all pixels to 'off'

// Initialise the humidity sensor board
pinMode(humidity_sensor_vcc, OUTPUT);
digitalWrite(humidity_sensor_vcc, LOW);

// Initialise the LUX-sensor
// Initialise the sensor
if(!tsl.begin())
{
    // Problem detecting the TSL2561
    Serial.print("No TSL2561 detected");
    while(1);
}

// Display some basic information on this sensor
displaySensorDetails();
// Setup the sensor gain and integration time
configureSensor();

Serial.println("");
}

int read_humidity_sensor() {
    digitalWrite(humidity_sensor_vcc, HIGH);
    delay(250);
    int value = analogRead(humidity_sensor_pin);
    digitalWrite(humidity_sensor_vcc, LOW);
    int humidity = 1023 - value;

    // Super dry 0-50
    if(humidity <= 50){
        delay(500);
        leds.setPixelColor(4, 245, 60, 5); //orange
        leds.setPixelColor(5, 245, 60, 5); //orange
        leds.setPixelColor(6, 245, 60, 5); //orange
        leds.setPixelColor(7, 245, 60, 5); //orange
    }
    // Dry 50 - 150
    if(humidity > 50 && humidity < 150){
        delay(500);
        leds.setPixelColor(4, 245, 60, 5); //orange
        leds.setPixelColor(5, 245, 60, 5); //orange
        leds.setPixelColor(6, 190, 0, 250); // purple
        leds.setPixelColor(7,190, 0, 250); // purple
    }

    // Need water

```

```

if(humidity > 150 && humidity < 300){
  delay(500);
  leds.setPixelColor(4, 190, 0, 250);// purple
  leds.setPixelColor(5, 190, 0, 250);// purple
  leds.setPixelColor(6, 190, 0, 250); //purple
  leds.setPixelColor(7, 0, 0, 150);// blue
}

// humid 300-500
if(humidity > 300 && humidity < 500){
  delay(500);
  leds.setPixelColor(4, 190, 0, 250);// purple
  leds.setPixelColor(5, 190, 0, 250);//purple
  leds.setPixelColor(6, 0, 0, 150);// blue
  leds.setPixelColor(7, 0, 0, 150);// blue
}

// so watery 500-600
if(humidity > 500 && humidity < 600){
  delay(500);
  leds.setPixelColor(4, 190, 0, 250);// purple
  leds.setPixelColor(5, 0, 0, 150);// blue
  leds.setPixelColor(6, 0, 0, 150);; // blue
  leds.setPixelColor(7, 0, 0, 150);// blue
}

// Full 600-850
if(humidity > 600 && humidity < 850){
  delay(500);
  leds.setPixelColor(4, 0, 0, 150);// purple
  leds.setPixelColor(5, 0, 0, 150);// blue
  leds.setPixelColor(6, 0, 0, 150);; // blue
  leds.setPixelColor(7, 0, 0, 150);// blue
}

// submerged in water (MAX) 850+
if (humidity >= 850){
  delay(500);
  leds.setPixelColor(4, 255, 0, 0);// Red
  leds.setPixelColor(5, 190, 0, 250);// purple
  leds.setPixelColor(6, 0, 0, 255);// blue
  leds.setPixelColor(7, 0, 0, 255);// blue
}

return 1023 - value;
}

int read_temperature() {
  //getting the voltage reading from the temperature sensor
  int reading = analogRead(tempPin);
  // converting that reading to voltage, for 3.3v arduino use
  ↪ 3.3
  float voltage = reading * 5.0;
  voltage /= 1024.0;

  // print out the voltage
  // Serial.print(voltage); Serial.println(" volts");

```

```

// now print out the temperature
float temperatureC = (voltage - 0.5) * 100 ; //converting
    ↪ from 10 mv per degree wit 500 mV offset
                                                    //to degrees
                                                    ↪ ((
                                                    ↪ voltage
                                                    ↪ - 500mV)
                                                    ↪ times
                                                    ↪ 100)

Serial.print(temperatureC);

if (temperatureC < 10.0){
    leds.setPixelColor(0, 0, 0, 255); //temp too low - blue
}

else if (temperatureC > 32.0){
    leds.setPixelColor(0, 255, 0, 0); //temp too high - red
}

else {
    leds.setPixelColor(0, 0, 255, 0); //good temp - green
}
}

void displaySensorDetails(void)
{
    sensor_t sensor;
    tsl.getSensor(&sensor);
    Serial.println("-----");
    Serial.print ("Sensor:      "); Serial.println(sensor.
    ↪ name);
    Serial.print ("Driver Ver:  "); Serial.println(sensor.
    ↪ version);
    Serial.print ("Unique ID:   "); Serial.println(sensor.
    ↪ sensor_id);
    Serial.print ("Max Value:   "); Serial.print(sensor.
    ↪ max_value); Serial.println(" lux");
    Serial.print ("Min Value:   "); Serial.print(sensor.
    ↪ min_value); Serial.println(" lux");
    Serial.print ("Resolution: "); Serial.print(sensor.
    ↪ resolution); Serial.println(" lux");
    Serial.println("-----");
    Serial.println("");
    delay(500);
}

//Configures the gain and integration time for the TSL2561

void configureSensor(void)
{
    // You can also manually set the gain or enable auto-gain
    ↪ support
    // tsl.setGain(TSL2561_GAIN_1X);      // No gain ... use in
    ↪ bright light to avoid sensor saturation
    // tsl.setGain(TSL2561_GAIN_16X);    // 16x gain ... use
    ↪ in low light to boost sensitivity

```



```

    tsl.enableAutoRange(true);           // Auto-gain ...
        ↪ switches automatically between 1x and 16x

// Changing the integration time gives you better sensor
    ↪ resolution (402ms = 16-bit data)
    tsl.setIntegrationTime(TSL2561_INTEGRATIONTIME_13MS);
        ↪ // fast but low resolution
//    tsl.setIntegrationTime(TSL2561_INTEGRATIONTIME_101MS);
        ↪ // medium resolution and speed
//    tsl.setIntegrationTime(TSL2561_INTEGRATIONTIME_402MS);
        ↪ // 16-bit data but slowest conversions

// Update these values depending on what you've set above!
Serial.println("-----");
Serial.print  ("Gain:          "); Serial.println("Auto");
Serial.print  ("Timing:       "); Serial.println("13 ms");
Serial.println("-----");
}

void loop() {

    leds.show(); // This sends the updated pixel color to the
        ↪ hardware.
    delay(delayval); // Delay for a period of time (in
        ↪ milliseconds).

// Get a new sensor event
    sensors_event_t event;
    tsl.getEvent(&event);

// Display the results (light is measured in lux)
    if (event.light)
    {
        Serial.print("Illuminance: ");
        Serial.print(event.light);
        Serial.println(" lux");
    }
    else
    {
        // If event.light = 0 lux, no reliable data from sensor (
            ↪ saturation)
        Serial.println("Sensor overload");
    }

    if (event.light < 10.0 && event.light != 0){
        leds.setPixelColor(1, 0, 0, 255); //not enough light/night
            ↪ time - blue
    }

    if (event.light > 10 && event.light < 100.0){
        leds.setPixelColor(1, 190, 0, 250); //not enough light -
            ↪ purple
    }

    if (event.light > 100 && event.light < 10000.0){
        leds.setPixelColor(1, 0, 250,0); //good shit - green
    }
}

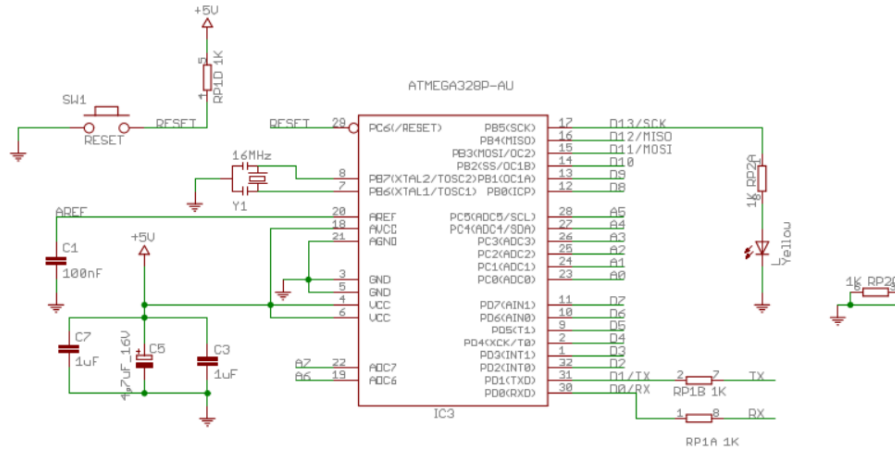
```

```
}  
  
if (event.light > 10000 && event.light <= 17000.0){  
    leds.setPixelColor(1, 250, 0,0); //too much - red  
}  
  
Serial.print("Humidity (0-1023): ");  
Serial.println(read_humidity_sensor());  
Serial.print("Temperature: ");  
Serial.print(read_temperature());  
Serial.println(" C");  
delay(1000);  
}
```

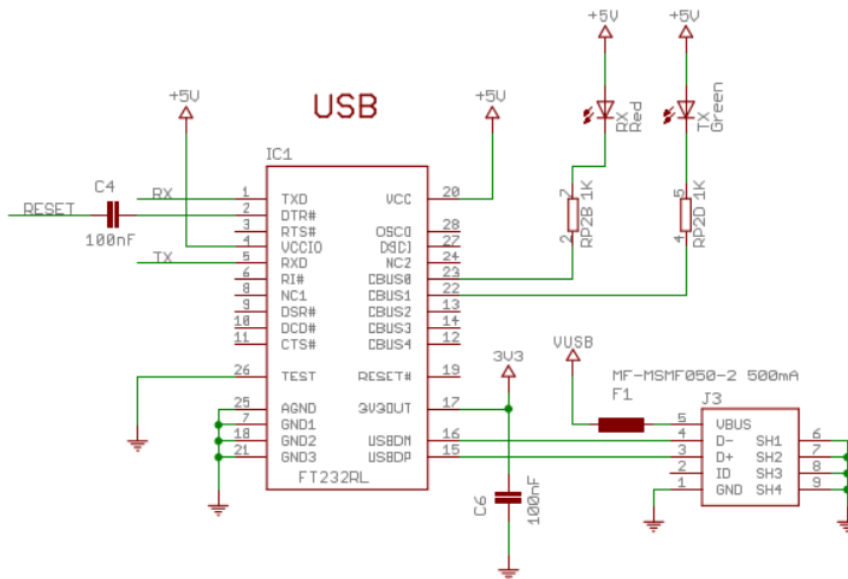
5 Skjemaer og Datasheets

5.1 Arduino Nano

Datasheet: <https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual123.pdf>



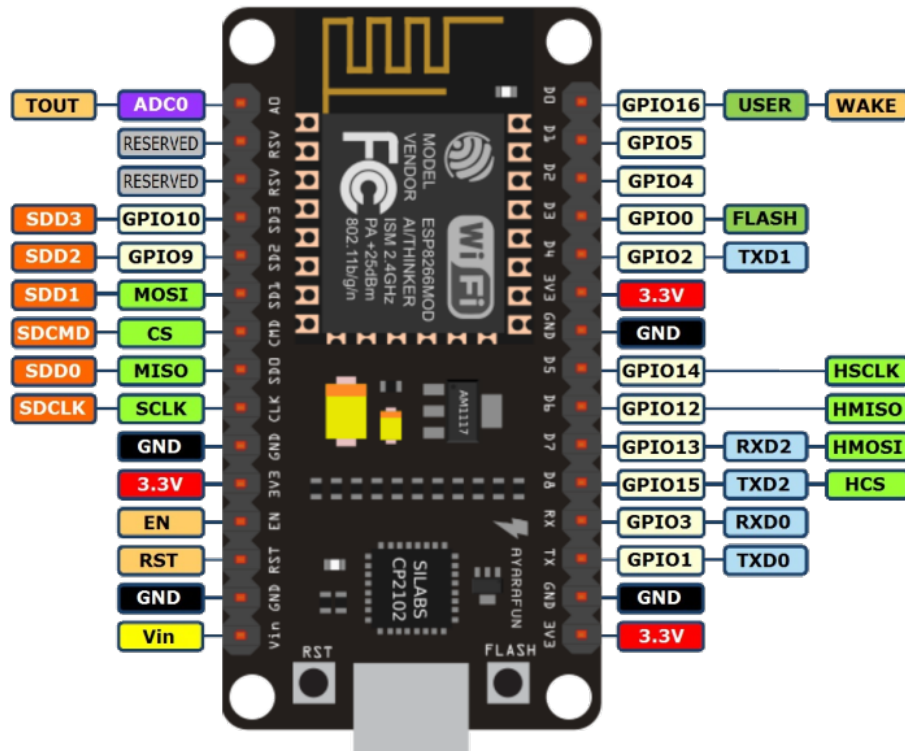
Figur 2: Skjema 1



Figur 3: Skjema 2

5.2 ESP8266

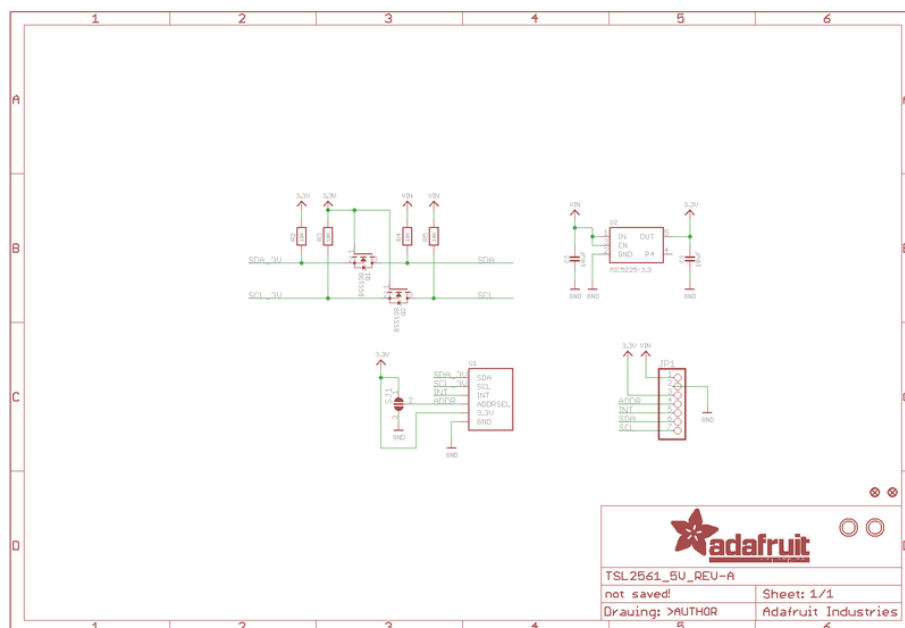
Datasheet: <http://download.arduino.org/products/UNOWIFI/0A-ESP8266-Datasheet-EN-v4.3.pdf>



Figur 4: ESP8266 PIN out

5.3 TSL2561

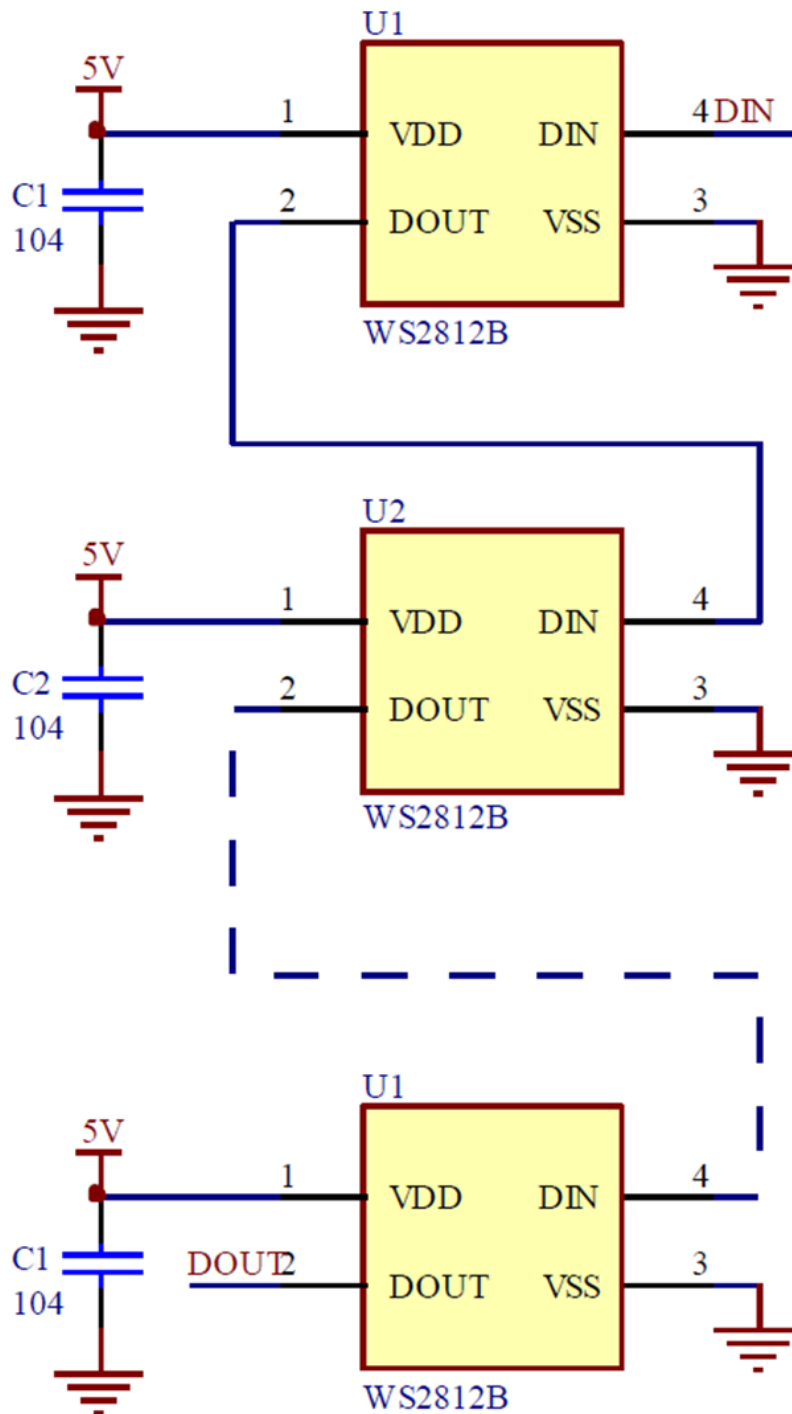
Datasheet: <https://cdn-shop.adafruit.com/datasheets/TSL2561.pdf>



Figur 5: TSL2561 Lux-sensor skjema

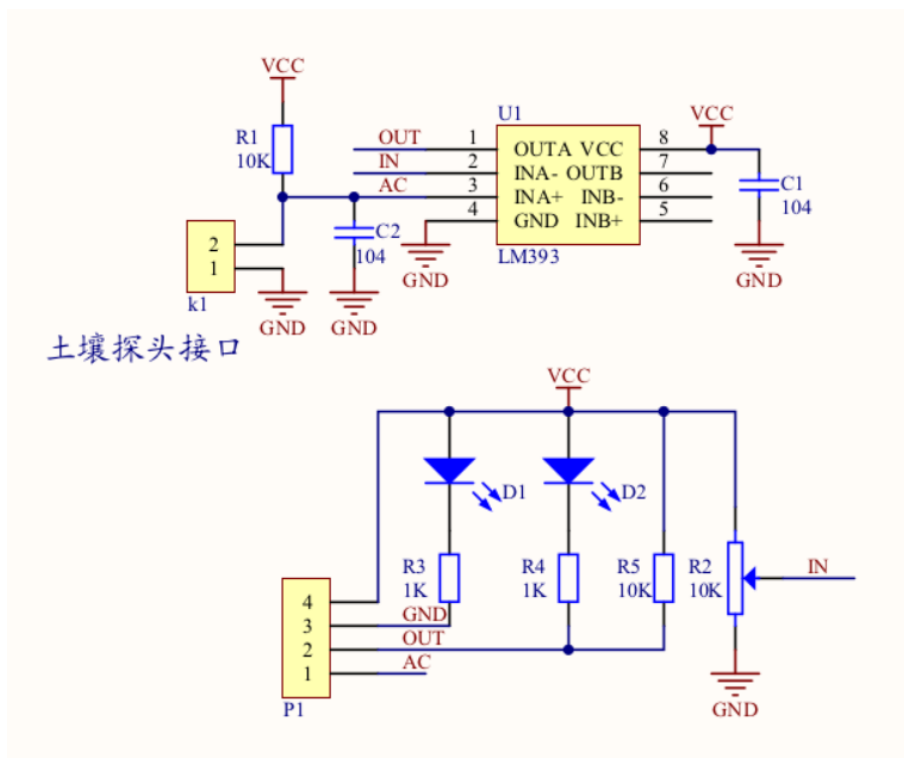
5.4 WS2812b

Datasheet: <https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf>



Figur 6: WS2812b

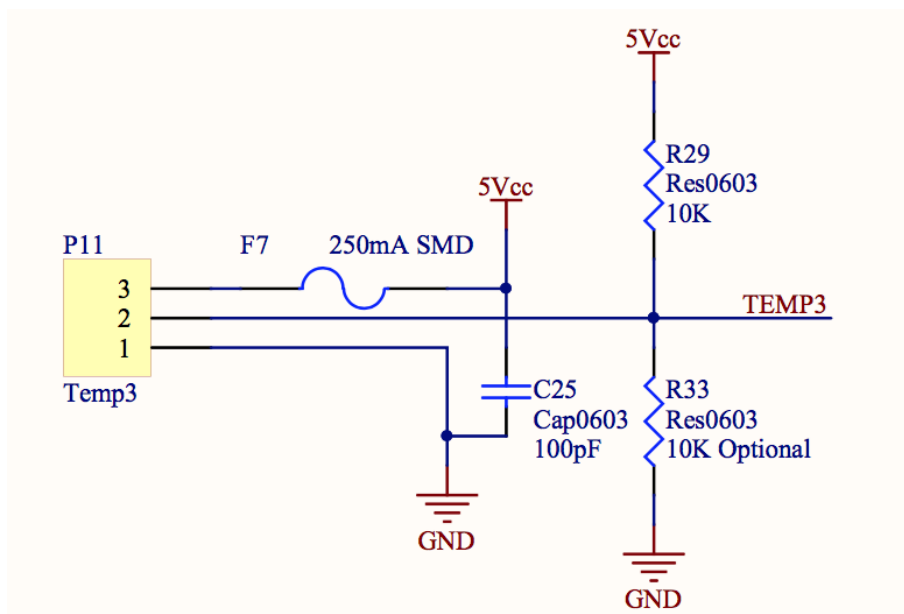
5.5 YL-69 og YL-38



Figur 7: YL69 og YL-38

5.6 TMP36(GZ)

Datasheet: http://www.analog.com/media/en/technical-documentation/data-sheets/TMP35_36_37.pdf



Figur 8: TMP36-sensor skjema