



# TTT4255 Elektronisk systemdesign, grunnkurs

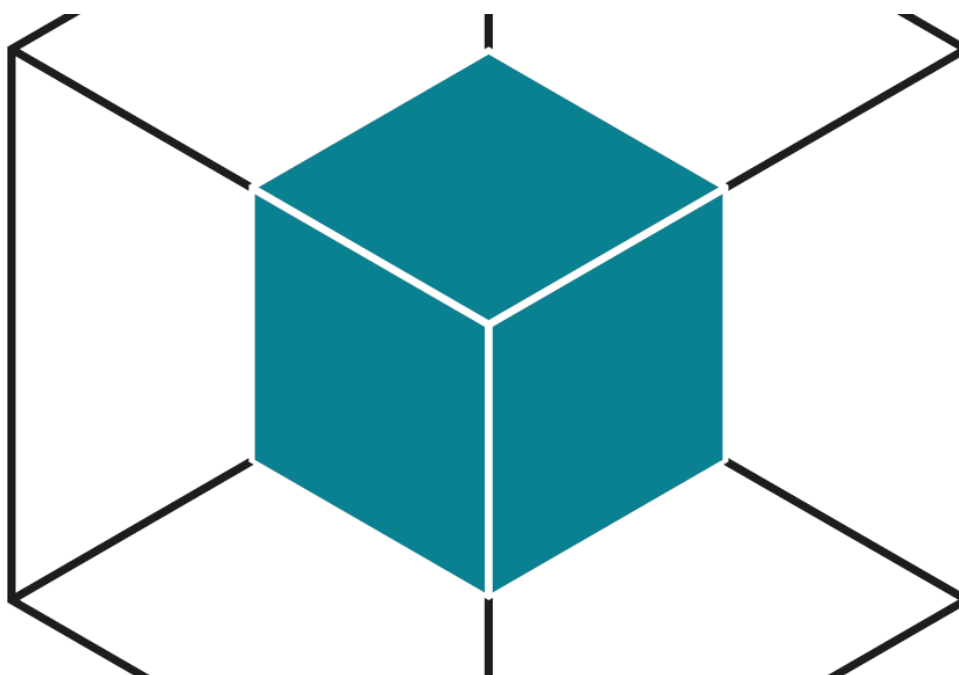
## K2: Styr LED over WiFi

Elektronisk systemdesign og innovasjon

---

Ida Bjørnevik, Sven Amberg, Amalie 28.06.2023  
Fridfeldt Hauge og Peter Magerøy

---



### Innhold

<b>Introduksjon</b>	<b>2</b>
Teori . . . . .	2
<b>Prosjektet</b>	<b>3</b>
Utstyrsliste . . . . .	3
Steg 1: Del nett . . . . .	3
Steg 2: Blynk . . . . .	5
Steg 3: Kode . . . . .	9
<b>Konklusjon</b>	<b>12</b>

---

## Introduksjon

Passer for deg med **ingen** forkunnskaper.

I denne modulen skal du bli kjent med Blynk appen og koble ESP32 til mobilen med WiFi for å styre en pinne.



**Figur 1:** Internettet.

### Teori for den interesserte

Internett, nettverk av datamaskiner som kommuniserer med hverandre i henhold til protokollen Internet Protocol (IP). I årenes løp har mange slike nettverk blitt bygget opp og knyttet sammen til ett stort, verdensomspennende nettverk. Selv om det også finnes isolerte nett som bruker den samme teknologien, er det dette globale nettverket som i dag kalles for internett.

Her er noen linker om internettets historie og hvordan det fungerer (ikke nødvendig for å gjennomføre oppgavene).

- <https://snl.no/internett>
- <https://xn--bredbnd-ixa.no/n%C3%A5r-kom-internett>

# Prosjektet

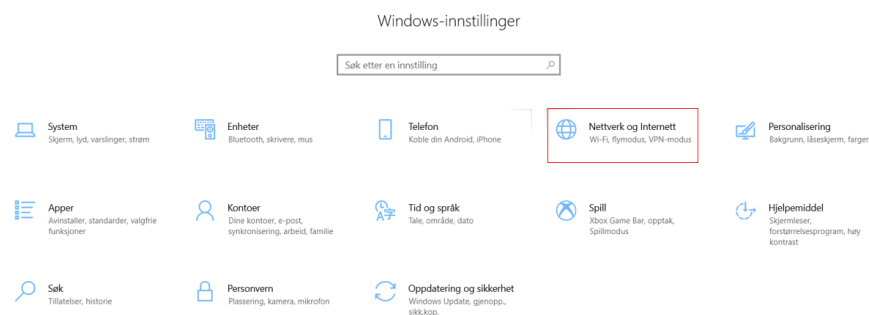
## Utstyrsliste

- WiFi
- ESP32
- LED

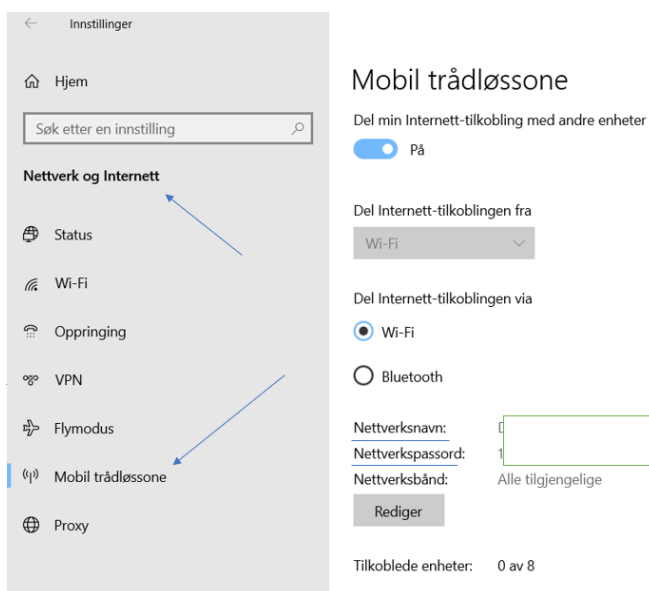
## Instruksjoner

### Steg 1: Del nett

Gå inn på Innstillinger > Internett > Trådløs sone. Rediger **Network band** til 2.4 GHz.



Figur 2: Innstillinger.



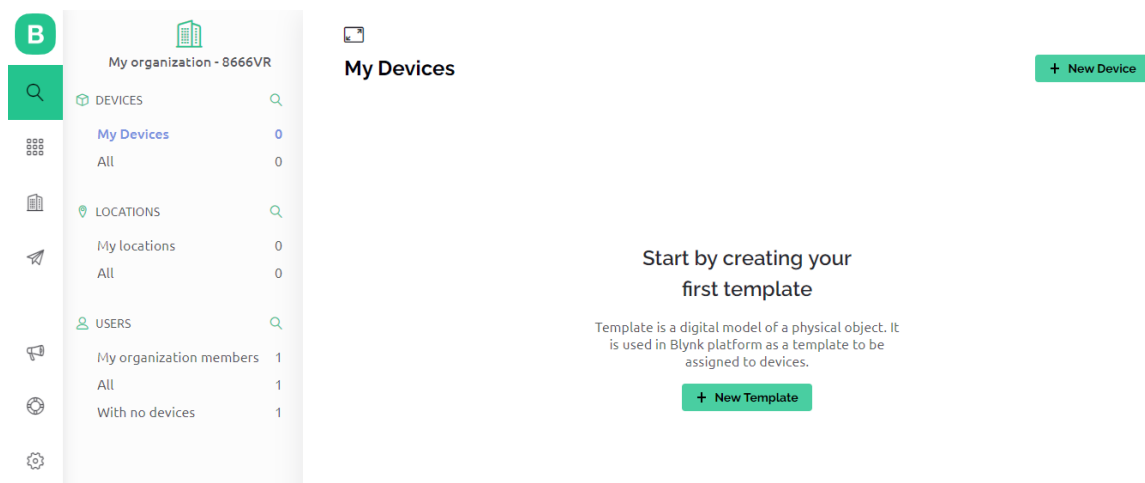
Figur 3: Trådløs sone.

Velg et nettverksnavn og passord, og start delingen av nett.

## Steg 2: Blynk

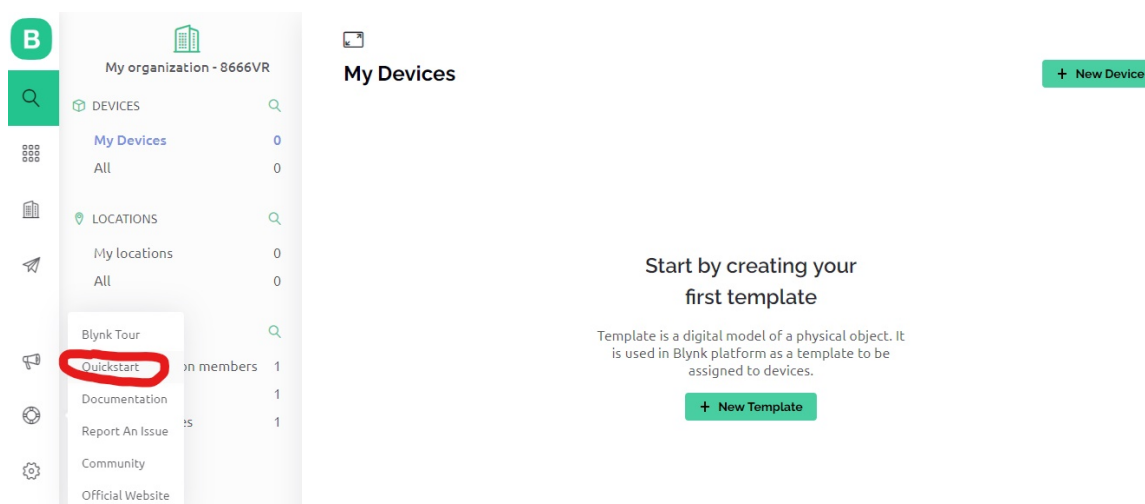
Vi skal nå begynne med å sette opp Blynk. Dette er en nettside som gjør det veldig enkelt å sette opp IOT-applikasjoner. Det er også en App du kan laste ned på mobilen din og koble til ESP-en fra. Vi skal derimot kun bruke Blynk i nettleseren i denne modulen. Du skal få se at Blynk kan automatisk generere mye av koden for oss.

Gå til <https://blynk.io/> og lag en bruker. Når du er inne skal det se ut som liknende i figur 4.



Figur 4: Innlogget i Blynk.

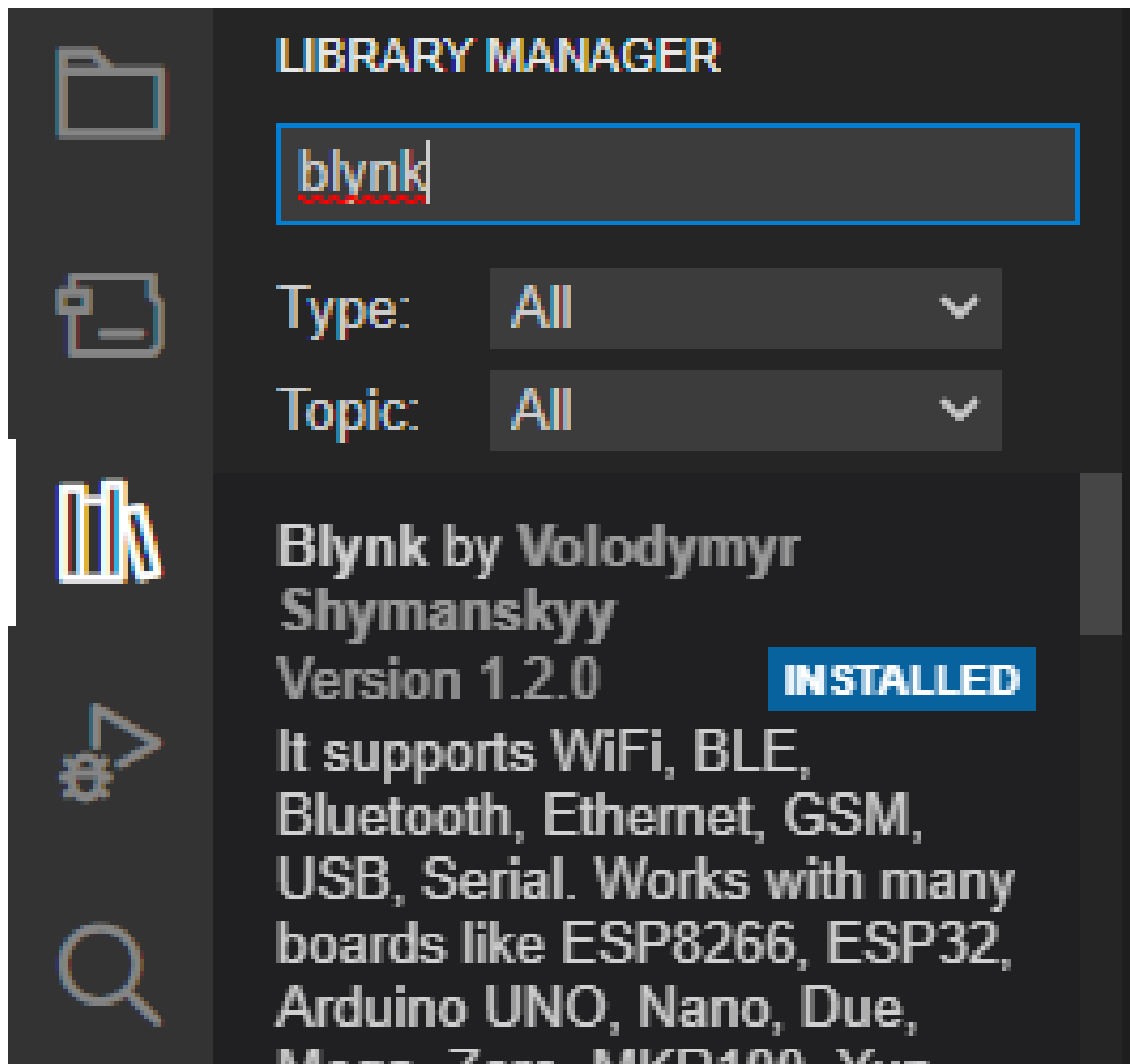
Herfra skal du navigere til **Quickstart**, vist i figur 5.



Figur 5: Quickstart.

Det dukker så opp et vindu. Trykk **Let's go!** og følg guiden. Under Hardware velger du **ESP32**. Under IDE velger du **Arduino**.

Gå over til Arduino IDE for å installere biblioteket til Blynk. Hvordan du installerer biblioteker er gått gjennom i modul I2. Det skal se ut slik vist i figur 6.



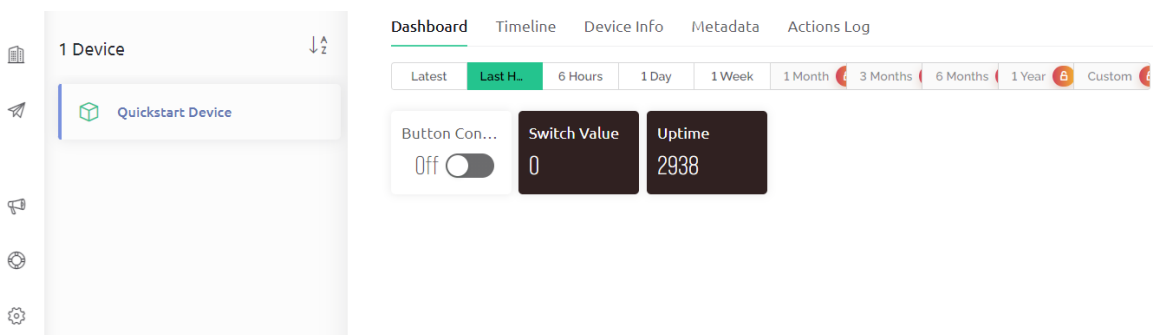
**Figur 6:** Blynk installert i Library Manager.

Etter å ha installert biblioteket til Blynk kan du gå tilbake til guiden hos Blynk. Under Code får du koden du skal bruke automatisk generert. Fyll inn nettverksnavn og passord før du kopierer koden. Det er viktig at dette er nettverksnavnet og passordet til nettet du deler fra PC-en.

**Figur 7:** Under Code får du automatisk generert kode for din mikrokontroller.

Legg merke til at det er tre viktige variabler her. Det er nettverksnavnet, passordet og et **BLYNK\_AUTH\_TOKEN**, som er strøket over i figur 7. Ditt token må være likt både i Arduino IDE og i Blynk for at du skal få koblet til. Når du har limt inn koden i Arduino IDE er du klar til å laste opp til ESP-en.

Trykk **Next** i quickstart guiden og enheten skal koble seg til. Når enheten er tilkoblet får du en liten feiring. Trykk så **Go To Device**. Du kommer nå til ditt dashboard. Det er herfra du skal styre din ESP. Du kan legge til widgeter som har forskjellig funksjonalitet. Det er allerede tre stykker lagt til i vårt dashboard; en bryter, og to klosser som informerer om bryterens tilstand og opptiden til enheten.




**Figur 8:** Dashboard med tre widgeter lagt til.


I ditt dashboard skal du nå trykke på de tre prikkene ved siden av der det står Quickstart Device. Velg **Edit Dashboard**. Hold musepekeren over bryterwidgeten og trykk på innstillingsikonet. Sjekk at alle verdier under Datastream er likt som i figur 9.

### Switch Settings ⓘ

TITLE (OPTIONAL)

**DataStream**  
 

ON VALUE

OFF VALUE  
 


☒ Show on/off labels

ON LABEL

OFF LABEL

LABEL POSITION

Button Control (V0)

Off 

Cancel

Save

**Figur 9:** Innstillinger til bryterwidget.

Gå ut av innstillingene og trykk **Save And Apply**. Alt i Blynk skal nå være satt opp slik vi ønsker. Bryteren vår bruker det som kalles en virtuell pinne på ESP-en til å kommunisere med den. I vårt tilfelle bruker vi virtuell pinne **V0**. Vi skal i neste steg bruke den virtuelle pinnen til å kontrollere en LED.



### Steg 3: Kode

Slik koden vi kopierte fra Blynk er nå vil ikke mikrokontrolleren gjøre noe spesielt når vi flipper bryteren vår i nettleseren. Dette skal vi gjøre noe med nå. Gå til prosjektet i Arduino IDE.

Vi skal begynne med å oppsett for en LED. Legg til for å definere en **LED** lik den GPIO-en vi tenker å koble til en LED, i dette eksempelet GPIO 2.

```
#define LED 2
```

**Figur 10:** Definerer GPIO.

I setup()-funksjonen legger vi til pinnemodus for LED-en.

```
pinMode(LED, OUTPUT);
```

**Figur 11:** Setter GPIO for LED til outputmodus.

Nå skal vi legge til slik at lyset blir slått på når vi har trykket inn bryteren på vårt dashboard i Blynk. Dette er kun en linje med kode. Legg til **digitalWrite(LED, value);** i funksjonen **BLYNK\_WRITE(V0)** slik vist under.

```
// This function is called every time the Virtual Pin 0 state
  changes
BLYNK_WRITE(V0)
{
  // Set incoming value from pin V0 to a variable
  int value = param.asInt();
  digitalWrite(LED, value);

  // Update state
  Blynk.virtualWrite(V1, value);
}
```

**Figur 12:** Setter output høyt når V0 er logisk høy.

Denne funksjonen kalles hver gang **V0** endrer tilstand, altså hver gang vi flipper bryteren. Her setter vi **value** til det heltallet vi tolker inputen fra bryteren som. Deretter kan vi skrive et digitalt signal til GPIO for å styre LED. Koble til en LED til den GPIO-en du definerte **LED** til i koden din. Du skal nå kunne kontrollere den fra dashboardet ditt i Blynk. Koden din skal nå se ut som den under.

```

/* Fill-in information from Blynk Device Info here */
#define BLYNK_TEMPLATE_ID          "TMPL4CybWTj_m"
#define BLYNK_TEMPLATE_NAME        "Quickstart Template"
#define BLYNK_AUTH_TOKEN           "<DITT_TOKEN>"

/* Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial

#define LED 2

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "<NETTVERKSNAVN>";
char pass[] = "<PASSORD>";

BlynkTimer timer;

// This function is called every time the Virtual Pin 0 state
// changes
BLYNK_WRITE(V0)
{
    // Set incoming value from pin V0 to a variable
    int value = param.asInt();
    digitalWrite(LED, value);

    // Update state
    Blynk.virtualWrite(V1, value);
}

// This function is called every time the device is connected
// to the Blynk.Cloud
BLYNK_CONNECTED()
{
    // Change Web Link Button message to "Congratulations!"
    Blynk.setProperty(V3, "offImageUrl", "https://static-image.
        nyc3.cdn.digitaloceanspaces.com/general/fte/
        congratulations.png");
    Blynk.setProperty(V3, "onImageUrl", "https://static-image.
        nyc3.cdn.digitaloceanspaces.com/general/fte/
        congratulations_pressed.png");
    Blynk.setProperty(V3, "url", "https://docs.blynk.io/en/
        getting-started/what-do-i-need-to-blynk/how-quickstart-
        device-was-made");
}

```

```

// This function sends Arduino's uptime every second to Virtual
// Pin 2.
void myTimerEvent()
{
    // You can send any value at any time.
    // Please don't send more than 10 values per second.
    Blynk.virtualWrite(V2, millis() / 1000);
}

void setup()
{
    // Debug console
    Serial.begin(115200);
    pinMode(LED, OUTPUT);

    Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
    // You can also specify server:
    //Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass, "blynk.cloud",
    //            80);
    //Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass, IPAddress
    //            (192,168,1,100), 8080);

    // Setup a function to be called every second
    timer.setInterval(1000L, myTimerEvent);
}

void loop()
{
    Blynk.run();
    timer.run();
    // You can inject your own code or combine it with other
    // sketches.
    // Check other examples on how to communicate with Blynk.
    // Remember
    // to avoid delay() function!
}

```

## Konklusjon

I denne modulen lærte du hvordan du setter opp Blynk og kobler ESP32 til Blynk med bare WiFi. Du fikk da til å skru av og på et LED lys.

### Refleksjonsspørsmål

- Hva skjer hvis du endrer knappen fra switch til push modus?
- Prøv å måle GPIO 2 og GND med multimeter, se på dens verdier når du skruer på LED lyset? Forklar hva som skjer og hvorfor.