

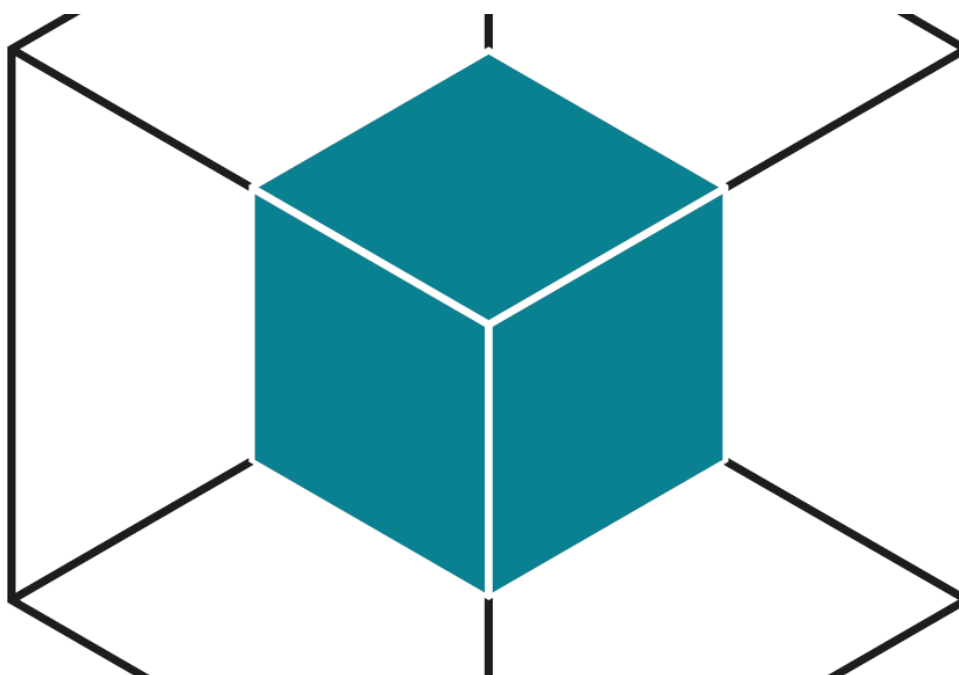


TTT4255 Elektronisk systemdesign, grunnkurs

K1: Koble til WiFi

Elektronisk systemdesign og innovasjon

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



Innhold

Introduksjon	2
Teori	2
Prosjektet	3
Utstyrliste	3
Steg 1: Inkludere bibliotek	3
Steg 2: Setup.	4
Konklusjon	7

Introduksjon

Passer for deg med **ingen forkunnskaper**. Det er en fordel å ha gjort modul **A1** og **K1**.

I denne modulen skal du bli kjent med WiFi, se figur 1, og skal lære hvordan man kobler ESP32 til ett WiFi nettverk.



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

Utstysrliste

- WiFi (Trådløs sone)
- ESP32

Kode

Steg 1: Inkludere bibliotek

Siden vi bare vil at ESP32 skal koble seg til WiFi så koder vi alt i setup()-funksjonen.

Åpne et nytt prosjekt i Arduino.

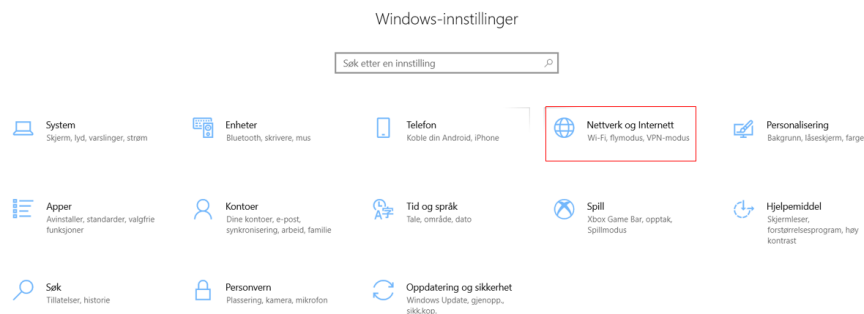
Først må vi inkludere wifi biblioteket.

```
#include "WiFi.h"
```

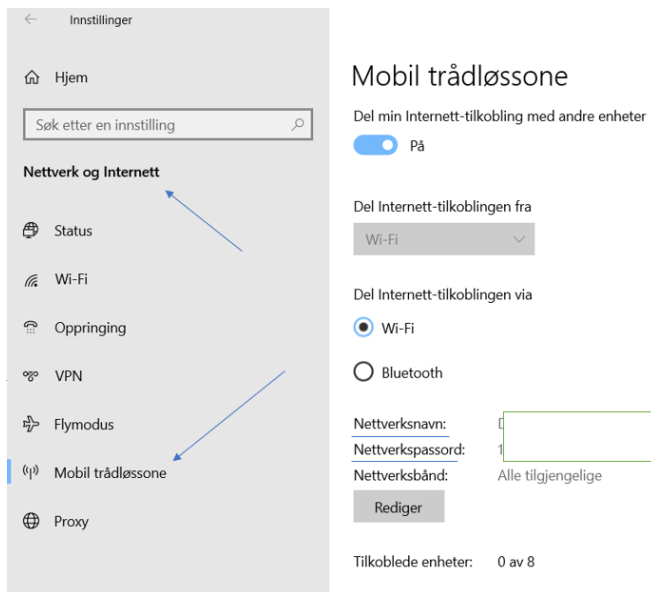
Figur 2: WiFi biblioteket

For å holde koden vår enkel å redigere, vil vi erklære to globale variabler, som skal holde både navnet på WiFi-nettverket vi vil koble til, og passordet. Siden vi ikke kan koble til Eduroam, må vi lage en trådløs sone med laptopen eller mobilen.

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

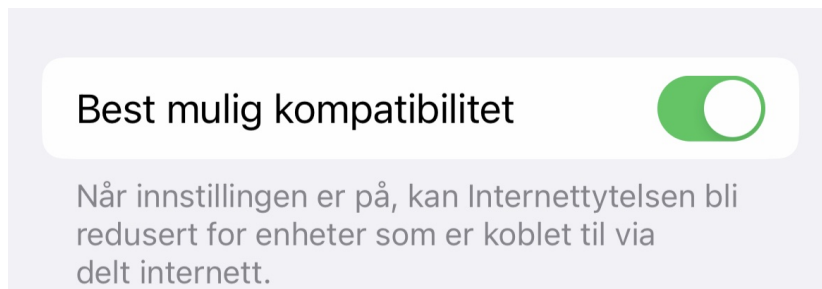


Figur 3: Innstillinger.



Figur 4: Trådløs sone.

Om du har Mac eller iPhone bør du velge alternativet **Best mulig kompatibilitet** som vist i figur 5. Dette gjør at nettet vi deler skjer over 2,4 GHz som er det vår ESP kan koble til.



Figur 5: For at nettverket skal være kompatibelt med ESP.

De to globale variablene vil se slik ut. Bare skriv inn navnet på ditt WiFi og passord inne i "....".

```
const char* ssid = "KarlSinWiFi";
const char* password = "12345678";
```

Figur 6: SSID og passord til ditt delte nett.

Steg 2: Setup.

Nå vil vi spesifisere setup()-funksjonen, hvor vi faktisk vil koble til WiFi-nettverket. Men først åpner vi en seriell tilkobling, slik at vi kan sende resultatet av programmet til PC.

Deretter kaller vi startmetoden på WiFi-objektet, og setter inn argumentene SSID (nettverksnavn) og passord. Dette starter forbindelsen til nettverket.

```
void setup() {  
  Serial.begin(115200);  
  WiFi.begin(ssid, password);  
}
```

Figur 7: Starte seriell og WiFi.

Etter det vil vi kjøre en while løkke til forbindelsen er etablert. For å gjøre det kan vi bruke statusmetoden på WiFi-objektet og vente på at resultatet samsvarer med WL_CONNECTED-enum.

Den vil da be om status til WiFi og sjekke at statusen = koblet til. Er den ikke det, vil den fortsette i whileløkken. Den repeterer programdelen i whileløkken til ESP32 er koblet til WiFi du oppgir i variablene.

Mellom hver iterasjon innfører vi en liten forsinkelse for å unngå en konstant prøving. (Flere av begrepene kan du sikkert ikke. Du vil lære om de i ITGK.)

Den skriver også til seriell at den prøver å koble seg til WiFi.

```
while (WiFi.status() != WL_CONNECTED) {  
  delay(500);  
  Serial.println("Connecting to WiFi..");  
}
```

Figur 8: Kjører en while løkke for å se om den er koblet til og skriver det ut.

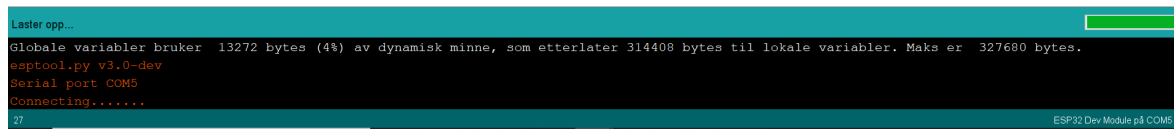
Ser programmet at det er koblet til WiFi, vil while løkken slutte og den vil skrive til seriell at den er koblet til.

```
Serial.println("Connected to the WiFi network");  
}
```

Figur 9: Skriver at den er koblet til.

Nå kan du laste opp programmet.

Husk å holde inne **BOOT** når det står **connecting** nederst i konsollen, vist i figur 10.

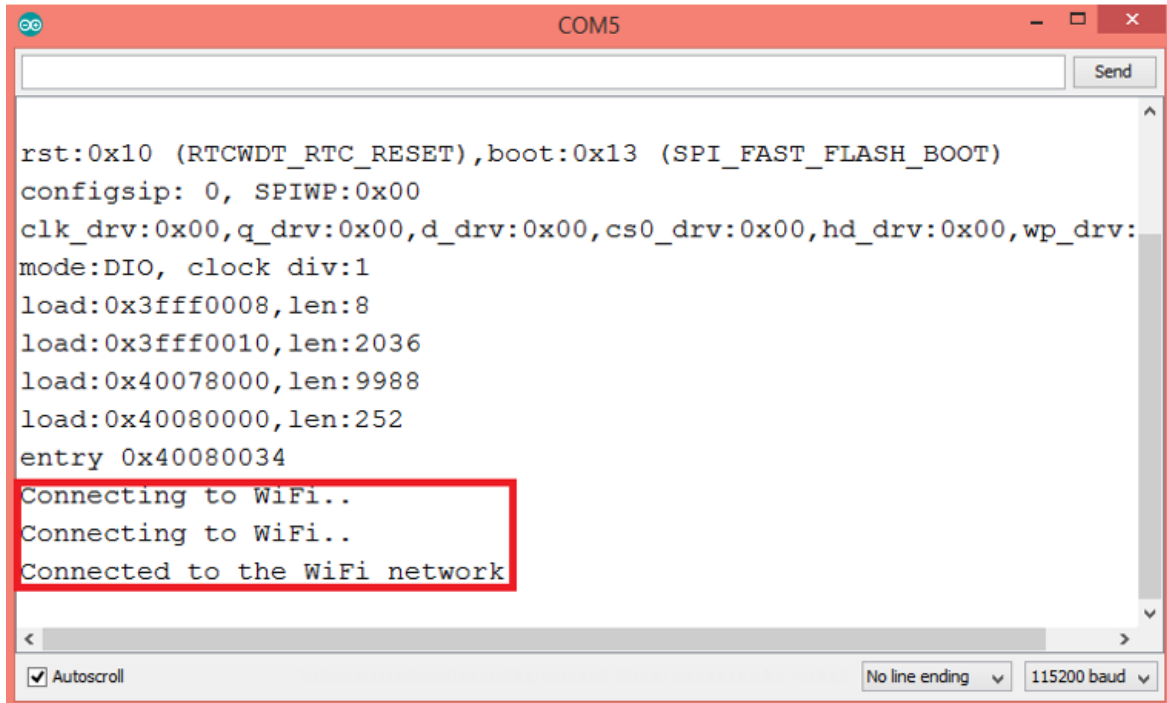


```
Laster opp...
Globale variabler bruker 13272 bytes (4%) av dynamisk minne, som etterlater 314408 bytes til lokale variabler. Maks er 327680 bytes.
esptool.py v3.0-dev
Serial port COM5
Connecting.....
27
```

ESP32 Dev Module på COM5

Figur 10: Sammenkobling mellom ESP32 og datamaskin.

Åpne seriell overvåkning og trykk på **RST**-knappen for å restarte ESP32. Da vil du se alt på skjermen fra starten av programmet. Du vil da få noe liknende som i figur 11.



```
rst:0x10 (RTCWDT_RTC_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
config: 0, SPIWP:0x00
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:
mode:DIO, clock div:1
load:0x3fff0008,len:8
load:0x3fff0010,len:2036
load:0x40078000,len:9988
load:0x40080000,len:252
entry 0x40080034
Connecting to WiFi..
Connecting to WiFi..
Connected to the WiFi network
```

Figur 11: Seriell overvåker når ESP32 kobler seg til WiFi.

Konklusjon

I denne modulen lærte du hvordan du setter opp WiFi for ESP32 og får den til å koble seg til WiFi.

Refleksjonsspørsmål

- Kan du tenke deg noen artige prosjekter der WiFi-tilkobling hadde vært nyttig?