

# Prototype - notat

Tittel: Iterasjon 1

Forfattere: Prosjekt - gruppe 2

Versjon: 1.0

Dato: 12.04.2024

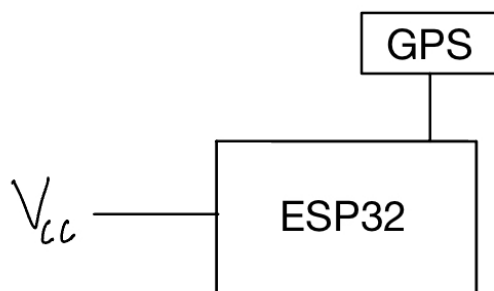
## Innhold

1	Introduksjon	1
2	Metode og testmiljø	2
3	Test resultat	3
4	Begrensninger og tiltak	6
5	Konklusjon	6
A	GPS-modul	7

---

## 1 Introduksjon

Vi tar for oss et system som vist i figur 1.



**Figur 1:** Prototype iterasjon 1

Figur 1 er en enkel framstilling av den første prototype iterasjonen. Den består av en ESP32 mikrokontroller samt en GPS-modul som er av typen *Adafruit Ultimate GPS* (A).

Krav til prototypen er at den skal kunne bruke GPS(global-positioning-system) til å finne lokasjonen til komponenten. Videre skal GPS-modulen kobles sammen med en ESP32 som tar inn dataen og omformer den til koordinater som deretter printes ut. Innebygd i GPS-modulen er en RTC(real-time-clock) som gir mulighet til enkel analyse av posisjon ved tid.

## 2 Metode og testmiljø

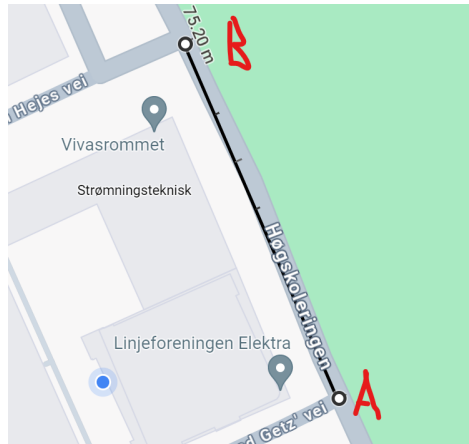
Vi bruker Arduino IDE plattformen til å kode mikrokontrolleren. Koden skal ta inn rå data fra GPS og omforme dette til koordinater som kan printes ut. Koden er vist i figur 2.

```
ESP32_GPS-print-location.ino
1 // Bibliotek
2 #include <Adafruit_GPS.h>
3
4 //Kobler GPS kommunikasjon til Serial2 på ESP32
5 //kan ha den på serial1 med mp ha RX1 og TX1 isåfall
6 #define GPSSerial Serial2
7
8 //variabel for å få ut GPS til serial kommunikasjon
9 Adafruit_GPS GPS(&GPSSerial);
10
11 void setup() {
12   Serial.begin(115200);
13   // Starter GPS ved 9600 baud
14   GPS.begin(9600);
15
16   /* Denne kommandoen gjør at vi både får sendt ut RMC som står
17    for (Recommended Minimum Navigation Information) samt GGA som står
18    for (Global Positioning System Fix data) som NMEA - setninger */
19   GPS.sendCommand(PMTK_SET_NMEA_OUTPUT_RMCGGA);
20
21   // Dette bestemmer oppdateringsfrekvensen til GPS'en
22   GPS.sendCommand(PMTK_SET_NMEA_UPDATE_1HZ); // 1 Hz update rate
23 }
24
25 void loop() {
26   /* Denne while setningen sjekker om det er kommet en ny
27    NMEA setning med informasjon som skal printes ut */
28   while (!GPS.newNMEAreceived()) {
29     GPS.read(); // Denne leser av for å se om det kommer noe
30   }
31
32   /* henter ut den NMEA informasjonen som ble sist lagret på GPS'en
33    og deretter parser denne, dette vil si å analysere NMEA informasjonen
34    som er lagret og trekke ut den informasjonen vi ønsker å printe */
35   if (GPS.parse(GPS.lastNMEA())) {
36     if (GPS.fix) { // Sjekker om GPS har en "fix", altså en posisjon å logge
37       // printe ut informasjonen som er sendt:
38       Serial.print("Location: ");
39       Serial.print(GPS.latitude, 4); Serial.print(GPS.lat);
40       Serial.print(", ");
41       Serial.print(GPS.longitude, 4); Serial.println(GPS.lon);
42     }
43   }
44 }
```

Figur 2: Kode for iterasjon 1

Vi henter først biblioteket til trackeren og definerer denne, samt en kommando som gjør det mulig å få ut GPS til serialmonitoren. I setup starter vi GPS ved 9600 baud, sammen med en kommando som sender RMC (Recommended Minimum Navigation Information) og GGA (Global Positioning System Fix data) som NMEA setninger og en oppdateringsfrekvens på 1Hz. I loopen er det en funksjon som sjekker om det er kommet en ny NMEA setning som skal printes. Neste er en if-setning som henter ut den NMEA informasjonen som sist ble lagret på GPS'en og parser denne informasjonen. Dette vil analysere NMEA informasjonen som er lagret og kun printe den informasjonen vi er interessert i.

For å teste prototypen settes den i en bil og kjører den langs en strekning som vist i figur 3.



**Figur 3:** Strekning som kjøres ved testing.

Mens vi kjører er prototypen slått på og den printer ut koordinater samt klokkeslett til hver måling. Vi starter i punkt A og slutter i punkt B og markerer tiden ved A og B. Etter vi har kjørt, analyserer vi dataen for å bestemme hvor nøyaktig den målte.

### 3 Test resultat

Etter GPS-modulen og ESP32 var koblet sammen, brukes plattformen Arduino IDE til å programmere mikrokontrolleren. Den rå dataen som blir sendt fra GPS-modulen blir først printet ut i seriel monitoren som vist i figur 4.

```

Serial Monitor x Output
Not connected. Select a board and a port to connect automatically

11:09:55.969 -> $GPGSA,A,2,25,06,20,,,,,,,,,2.82,2.63,1.00*0E
11:09:55.969 -> $GPGSV,2,1,05,25,63,131,28,06,19,031,28,20,09,090,26,11,,,29*4F
11:09:56.038 -> $GPGSV,2,2,05,12,,,27*7A
11:09:56.115 -> $GPRMC,090957.000,A,6325.0981,N,01024.4125,E,0.36,0.00,110424,,,A*6C
11:09:56.888 -> $GPGGA,090958.000,6325.0981,N,01024.4109,E,1,03,2.63,108.6,M,41.5,M,,*63
11:09:56.973 -> $GPGSA,A,2,25,06,20,,,,,,,,,2.82,2.63,1.00*0E
11:09:56.973 -> $GPRMC,090958.000,A,6325.0981,N,01024.4109,E,0.51,0.00,110424,,,A*6C
11:09:57.886 -> $GPGGA,090959.000,6325.0981,N,01024.4096,E,1,03,2.63,108.6,M,41.5,M,,*65
11:09:57.953 -> $GPGSA,A,2,25,06,20,,,,,,,,,2.82,2.63,1.00*0E
11:09:57.953 -> $GPRMC,090959.000,A,6325.0981,N,01024.4096,E,0.49,0.00,110424,,,A*63
11:09:58.907 -> $GPGGA,091000.000,6325.0980,N,01024.4090,E,1,03,2.64,108.6,M,41.5,M,,*61
11:09:58.953 -> $GPGSA,A,2,25,06,20,,,,,,,,,2.82,2.64,1.00*09
11:09:58.953 -> $GPRMC,091000.000,A,6325.0980,N,01024.4090,E,0.54,0.00,110424,,,A*6C
11:09:59.129 -> $GPGGA,091000.202,6325.0970,N,01024.4010,E,1,04,2.53,106.4,M,41.5,M,,*69
11:09:59.223 -> $GPGSA,A,3,25,06,11,20,,,,,,,,,2.71,2.53,0.96*0E
11:09:59.223 -> $GPRMC,091000.202,A,6325.0970,N,01024.4010,E,0.59,274.06,110424,,,A*61
11:10:00.906 -> $GPGGA,091002.000,6325.0839,N,01024.3342,E,1,04,2.53,48.6,M,41.5,M,,*5D
11:10:00.973 -> $GPGSA,A,3,25,06,11,20,,,,,,,,,2.71,2.53,0.96*0E
11:10:00.973 -> $GPGSV,2,1,06,25,63,131,27,11,38,067,30,06,19,031,27,37,17,167,*7A
11:10:01.041 -> $GPGSV,2,2,06,20,09,090,26,12,,,24*4C
11:10:01.127 -> $GPRMC,091002.000,A,6325.0839,N,01024.3342,E,1.81,78.82,110424,,,A*5A

```

**Figur 4:** Rå data fra GPS

Når koden fra figur (2) implementeres får man ut disse verdiene:

```
Output Serial Monitor X
Not connected. Select a board and a port to connect automatically.
11:07:01.023 -> Location: 6325.0894N, 1024.3778E
11:07:01.071 -> Location: 6325.0894N, 1024.3778E
11:07:01.979 -> Location: 6325.0898N, 1024.3774E
11:07:02.094 -> Location: 6325.0898N, 1024.3774E
11:07:02.988 -> Location: 6325.0894N, 1024.3752E
11:07:03.067 -> Location: 6325.0894N, 1024.3752E
11:07:04.004 -> Location: 6325.0889N, 1024.3741E
11:07:04.087 -> Location: 6325.0889N, 1024.3741E
11:07:05.009 -> Location: 6325.0884N, 1024.3732E
11:07:05.045 -> Location: 6325.0884N, 1024.3732E
11:07:06.001 -> Location: 6325.0889N, 1024.3724E
11:07:06.070 -> Location: 6325.0889N, 1024.3724E
11:07:07.001 -> Location: 6325.0889N, 1024.3718E
11:07:07.073 -> Location: 6325.0889N, 1024.3718E
11:07:08.005 -> Location: 6325.0889N, 1024.3710E
11:07:08.069 -> Location: 6325.0889N, 1024.3710E
11:07:08.986 -> Location: 6325.0889N, 1024.3706E
```

**Figur 5:** Lokasjon printet til serial monitor

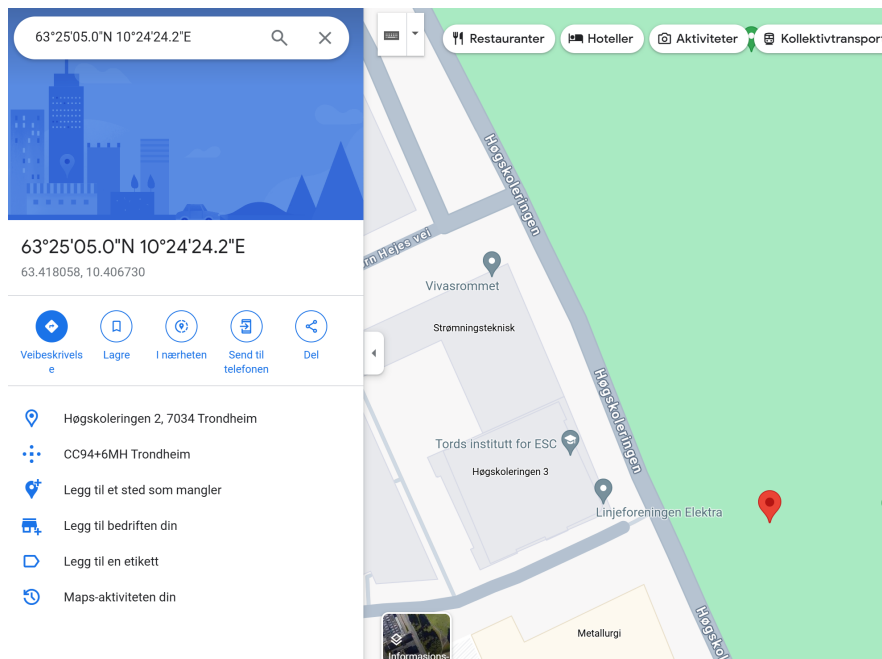
Dette gir oss en mye bedre oversikt over hva dataen vi får ut faktisk forteller oss om posisjonen. Denne dataen kan vi bruke til å fastsette posisjonen til GPS-modulen presist. Resultatene til posisjon A og B presenteres med henholdsvis koordinater, google maps plotting av koordinater og tid som vist i figur 4.

Vi startet gps-en her:

```
0:39:15.450 -> Lokasjon: 6325.0835, 1024.4038.
0:39:15.485 -> Lokasjon: 6325.0835, 1024.4038.
0:39:16.356 -> Lokasjon: 6325.0835, 1024.4041.
0:39:16.448 -> Lokasjon: 6325.0835, 1024.4041.
0:39:16.526 -> Lokasjon: 6325.0835, 1024.4041.
0:39:17.354 -> Lokasjon: 6325.0835, 1024.4038.
0:39:17.431 -> Lokasjon: 6325.0835, 1024.4038.
0:39:17.647 -> Lokasjon: 6325.0835, 1024.4038.
```

**Figur 6:** Lokasjonsdata fra Serial monitoren ved start

Lokasjon før skrevet inn i google maps:



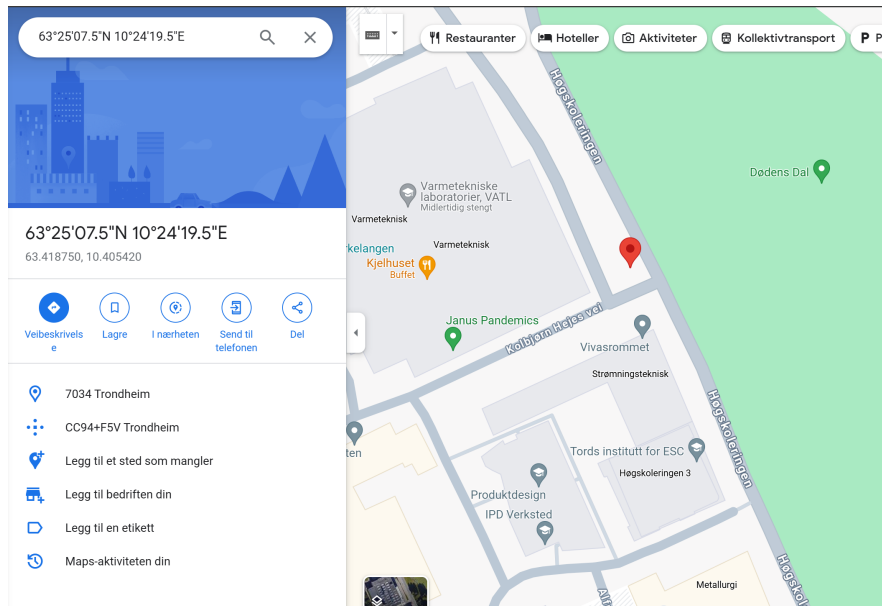
**Figur 7:** Koordinater start skrevet inn i google maps

Lokasjon etter:

```
.478 -> Lokasjon: 6325.1250, 1024.3271.
.556 -> Lokasjon: 6325.1250, 1024.3271.
.403 -> Lokasjon: 6325.1250, 1024.3264.
.489 -> Lokasjon: 6325.1250, 1024.3264.
.534 -> Lokasjon: 6325.1250, 1024.3264.
.448 -> Lokasjon: 6325.1255, 1024.3252.
.480 -> Lokasjon: 6325.1255, 1024.3252.
.741 -> Lokasjon: 6325.1255, 1024.3252.
```

**Figur 8:** Lokasjonsdata fra Serial monitoren ved slutt

Lokasjon etter skrevet inn i google maps:



**Figur 9:** Koordinater slutt skrevet inn i google maps

Her kan vi se at posisjonen til gps modulen er ganske nøyaktig fra hvor vi var.

## 4 Begrensninger og tiltak

Begrensinger til prototype 1:

- Presisjonen til GPS-modulen. (+- 5 meter)
- Avvik ved blokkert tilsyn til åpen himmel.
- Manuelt plotting av posisjon

Eventuelle tiltak:

- Koble GPS-modul til antennen i bilen
- Automatisk sending av data fra seriel monitor til kart som displayer posisjon.

## 5 Konklusjon

GPS-modulen og ESP32 greide å kommunisere sammen, i tillegg fikk vi printet ut lokasjon. Avviket på posisjonen virker å være på ca +- 5 meter. Disse avvikende kan komme av hardwarens begrenset nøyaktighet, GPS-avvik under tak og mulige avvik i google maps plotting. Disse avvikende kan forbedres med eventuelle tiltak.

## A GPS-modul

Du kan finne mer informasjon om GPS-modulen levert av Adafruit på: [Adafruit-nettsiden](#).