

Prototype - notat

Tittel: Iterasjon 3

Forfattere: Prosjekt - gruppe 2

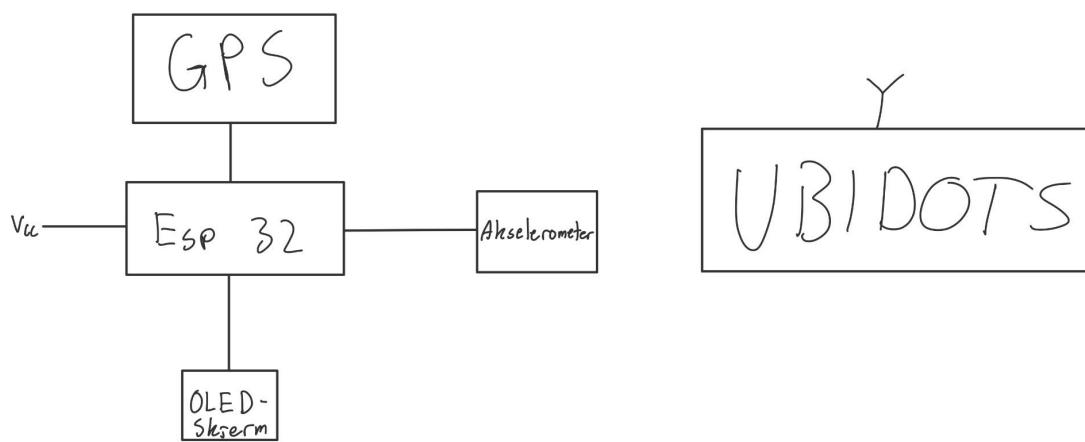
Versjon: 1.0 Dato: 19.04.2024

Innhold

1	Introduksjon	1
2	Metode og testmiljø	3
3	Test resultat og diskusjon	6
4	Begrensninger og tiltak	7
5	Konklusjon	8

1 Introduksjon

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



Figur 1: Prototype iterasjon 3

I denne iterasjon implementerte vi en løsning som gjør det mulig å få displayet dataen vi

henter ut, dette på IOT - nettsiden Ubidots. Selve designet besto av en ESP32, GPS - modul, akselerometer og en OLED skjerm som viste verdien for akselerasjonen. Tidligere har informasjonen vi ønsket blitt printet ut til serial monitoren, derav lokasjonen. Denne nye løsningen gjør det mulig å få displayet lokasjonen uten av å være nødt til å manuelt taste dette inn, istedenfor blir dette sendt til nettsiden som plotter dette inn i et kart for oss.

2 Metode og testmiljø

Koden vi benyttet ligger under, denne er da delt i 2 deler, der ene delen er en header fil for "setup" funksjoner, og den andre er hovedkoden. Denne header delen ligger under:

```
ESP32_akselerometer_GPS_ubidots.publish ino setupFunctions.h
1  /* Header fil som inneholder alle
2  de ulike setup - funksjonene */
3
4  #include <Adafruit_GPS.h>
5  #include <Arduino.h>
6  #include <ubidotsesp32MQTT.h>
7  #include "GY521.h"
8
9  #define GPSserial Serial2
10 Adafruit_GPS GPS(&GPSSerial);
11
12 GY521 sensor(0x68); //I2C port for kommunikasjon
13
14 // Ubidots credentials
15 const char *UBIDOTS_TOKEN = "BBUS-pE9omf3hxLaL6uzGCVIMoE27WVVTSX";
16 const char *WIFI_SSID = "NTNU-IOT";
17 const char *WIFI_PASS = "";
18
19 // GPS ubidots labels
20 const char *DEVICE_LABEL_gps = "gps_position";
21 const char *VARIABLE_gps = "gps";
22 const char *VARIABLE_satelitt = "satelitt";
23 const char *VARIABLE_satelitt_kvalitet = "kvalitet";
24
25 //akselerometer ubidots labels
26 const char *DEVICE_LABEL_akselerometer= "akselerometer";
27 const char *VARIABLE_akselerometer_verdi = "aks_verdi";
28
29 Ubidots ubidots(UBIDOTS_TOKEN);
30
31
32 void ubidots_setup()
33 {
34     // ubidots connection
35     ubidots.setup();
36     ubidots.reconnect();
37     ubidots.subscribeLastValue(DEVICE_LABEL_akselerometer, VARIABLE_akselerometer_verdi);
38     ubidots.subscribeLastValue(DEVICE_LABEL_gps, VARIABLE_gps);
39     ubidots.subscribeLastValue(DEVICE_LABEL_gps, VARIABLE_satelitt);
40     ubidots.subscribeLastValue(DEVICE_LABEL_gps, VARIABLE_satelitt_kvalitet);
41 }
42
43 void internet_connection()
44 {
45     // koble til internettet
46     Serial.println("Initializing Wi-Fi..");
47     ubidots.connectToWiFi(WIFI_SSID, WIFI_PASS);
48     if (ubidots.connected()) {
49         Serial.println("Wi-fi connected!");
50     } else {
51         Serial.println("Failed to connect to Wi-Fi!");
52     }
53 }
54
55 // Funksjon for å kalibrere akselerometret
56 void calibrate_accelerometer() {
57     Serial.println("Kalibrerer akselerometer");
58     int resolution = 100;
59     float calx = 0, caly = 0, calz = 0;
60
61     for (int i = 0; i < resolution; i++) {
62         sensor.read();
63         calx += sensor.getAccelX();
64         caly += sensor.getAccelY();
65         calz += sensor.getAccelZ();
66     }
67 }
68
69 sensor.axe = -calx / resolution;
70 sensor.aye = -caly / resolution;
71 sensor.aze = -calz / resolution;
72 Serial.println("Akselerometer er ferdig kalibrert");
73 }
74
75
76 // Funksjon for å initialisere GPS
77 void initialize_GPS() {
78     GPS.begin(9600);
79     GPS.sendCommand(PMTK_SET_NMEA_OUTPUT_RMCGGA);
80     GPS.sendCommand(PMTK_SET_NMEA_UPDATE_1HZ);
81     Serial.println("GPS er ferdig initialisert");
82 }
83
84 // Funksjon for å initialisere akselerometret
85 void initialize_sensor() {
86     delay(100);
87     while (!sensor.wakeup()) {
88         Serial.println("Kan ikke koble til GY521: sjekk GY521 adresse (0x68/0x69)");
89     }
90     sensor.setAccelSensitivity(0);
91     sensor.setThrottle();
92     Serial.println("Akselerometer er ferdig initialisert");
93 }
94 }
```

Hovedkoden så slik ut:

```
ESP32_akselerometer_GPS_ubidots-publish.ino      setupFunctions.h
1  #include "setupFunctions.h"
2
3  void setup()
4  {
5    Wire.begin();
6    Serial.begin(115200);
7    initialize_GPS();
8    initialize_sensor();
9    calibrate_accelerometer();
10   internet_connection();
11   ubidots_setup();
12 }
13
14 /* Denne lagrer "context" som dynamisk minne på
15 størpelse på 50 bytes. "malloc" reserverer minnet
16 til heapen som er minne som en kan bruke under kjøring
17 samt er dynamisk som gjør det mulig å endre verdien under
18 kjøring av programmet */
19 char* context = (char*)malloc(sizeof(char) * 50);
20
21 /* Konverterer koordinatene fra grader og minutter om til
22 desimal grader */
23 double konverterTilDesimalgrader(double koordinat) {
24   int grader = static_cast<int>(koordinat) / 100;
25   double desimalMinutter = fmod(koordinat, 100.0);
26   return grader + (desimalMinutter / 60.0);
27 }
28
29 void ubidots_connection()
30 {
31   //Om forbindelsen mistes koble til på nytt
32   if (!ubidots.connected())
33   {
34     ubidots.reconnect();
35     ubidots.subscribeLastValue(DEVICE_LABEL_akselrometer, VARIABLE_akselrometer_verdi);
36     ubidots.subscribeLastValue(DEVICE_LABEL_gps, VARIABLE_gps);
37     ubidots.subscribeLastValue(DEVICE_LABEL_gps, VARIABLE_satellitt);
38     ubidots.subscribeLastValue(DEVICE_LABEL_gps, VARIABLE_satellitt_kvalitet);
39   }
40 }
41
42 void gps_connection()
43 {
44   //Mye gps data mottatt og leses av
45   while (Gps.newNMEAReceived()) {
46     Gps.read();
47   }
48
49 //Hvis gps ikke finner posisjonen
50 if (Gps.fix == false)
51 {
52   Serial.println("Timer ikke posisjon");
53   delay(2500);
54 }
55
56 float acceleration_total(){ //Funksjon som henter ut akselerasjonsdata i x,y,z retning for å lage en akselerasjonsvektor som logger total akselerasjon
57   sensor.read();
58   float ax = sensor.getAccelX(); //Henter akselerasjon i x retning
59   float ay = sensor.getAccelY(); //Henter akselerasjon i y retning
60   float az = sensor.getAccelZ(); //Henter akselerasjon i z retning
61   float atot = sqrt((ax*ax)+(ay*ay)+(az*az)); //Akselerasjonsvektoren blir laget
62
63   /*Serial.print(" Akselerasjon din er "); //Printer ut akselerasjonsvektoren
64   Serial.print(atot);*/
65
66   Serial.println(" m/s^2");
67   return atot;
68 }
69
70 void ubidots_publish()
71 {
72   //dekrypterer NMEA - settningen
73   if (Gps.parse(Gps.lastNMEA)) {
74     // forteller oss om kvalitet, fra 1 - 6
75     Serial.print("Kvalitet: ");
76     Serial.println(Gps.fixquality);
77     delay(2500);
78
79   // sier hvor mange satellitter den er koblet til
80   Serial.print("Antall satellitter: ");
81   Serial.println((int)Gps.satellites);
82   delay(2500);
83
84   if (Gps.fix == true) {
85     // Konverterer grader og minutter til desimalgrader
86     double breddegrader = konverterTilDesimalgrader(Gps.latitude);
87     double lengdegrader = konverterTilDesimalgrader(Gps.longitude);
88
89     // Printer ut lokasjon i decimal grader
90     Serial.print(breddegrader, 8);
91     Serial.print(", ");
92     Serial.print(lengdegrader, 8);
93     Serial.println(",");
94
95     // Konverterer desimalgrader til strenger
96     char str_lat[20];
97     char str_lng[20];
98     sprintf(str_lat, "%f", breddegrader);
```

```

99     sprintf(str_lng, "%f", lengdegrader);
100
101    // legger til context som "lat" og "lng" fra
102    // breddegrad og lengdegrad
103    ubidots.addContext("lat", str_lat);
104    ubidots.addContext("lng", str_lng);
105
106    //kvalitet og antall satellitter
107    int satellitt = (int)gps.satellites;
108    int kvalitet = GPS.fixquality;
109
110    // sender context som er gps data til ubidots
111    ubidots.getContext(context);
112    ubidots.add(VARIABLE_gps, 1, context);
113    ubidots.add(VARIABLE_satellitt, satellitt);
114    ubidots.add(VARIABLE_satellitt_kvalitet, kvalitet);
115
116    // hente ut akselerasjon verdien
117    acceleration_total();
118    float akselerasjon = acceleration_total(); // Example temperature value
119    ubidots.add(VARIABLE_akselerometer_verdi, akselerasjon);
120
121    // legge inn knappetrykk her
122
123    ubidots.publish(DEVICE_LABEL_akselerometer);
124    ubidots.publish(DEVICE_LABEL_gps);
125
126    // forteller oss om data kom trygt frem til ubidots
127    int publishResult = ubidots.publish(DEVICE_LABEL_gps);
128    if (publishResult == 1) {
129        serial.println("Publisert på ubidots (gps)");
130    } else {
131        serial.print("Fikk ikke til å publisere (gps), feilkode: ");
132        Serial.println(publishResult);
133    }
134
135    // forteller oss om data kom trygt frem til ubidots
136    int publishResult_2 = ubidots.publish(DEVICE_LABEL_akselerometer);
137    if (publishResult_2 == 1) {
138        serial.println("Publisert på ubidots (akselerasjon)");
139    } else {
140        Serial.print("Fikk ikke til å publisere akselerasjon, feilkode: ");
141        Serial.println(publishResult_2);
142    }
143}
144
145
146 void loop()
147 {
148    ubidots_connection();
149    acceleration_total();
150    gps_connection();
151    ubidots.loop();
152    ubidots_publish();
153}

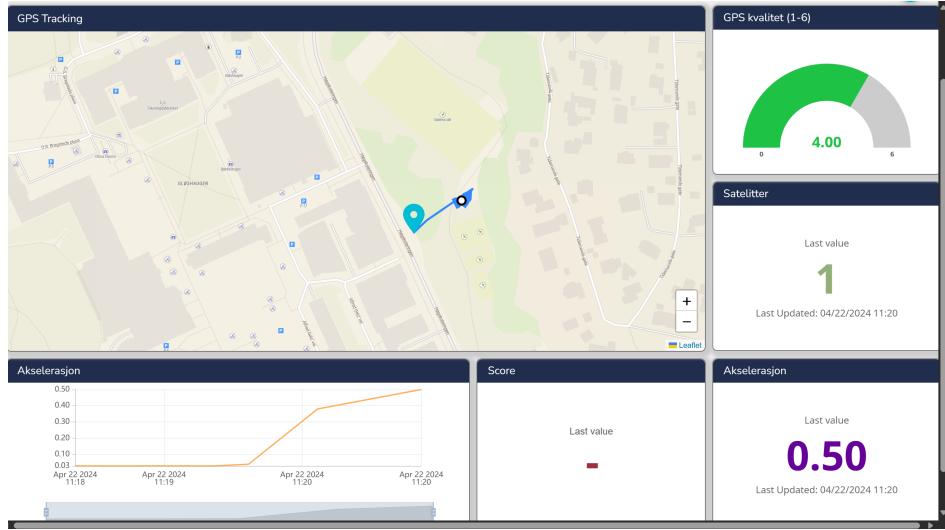
```

Figur 2: Hovedkode

Det nye i denne iterasjonen er muligheten til å sende dataen fra sensorene til nettsiden Ubidots som gir oss en oversiktlig måte å se dataen på. Koden prøver først å koble til Ubidots via internett. Når den klarer denne subscriberden på et ”emne”, her er ”emne” verdiene til sensorene. Etter dette gir den beskjed at den vellykket sendte data til Ubidots, hvis den ikke klarer det gir den også beskjed om dette.

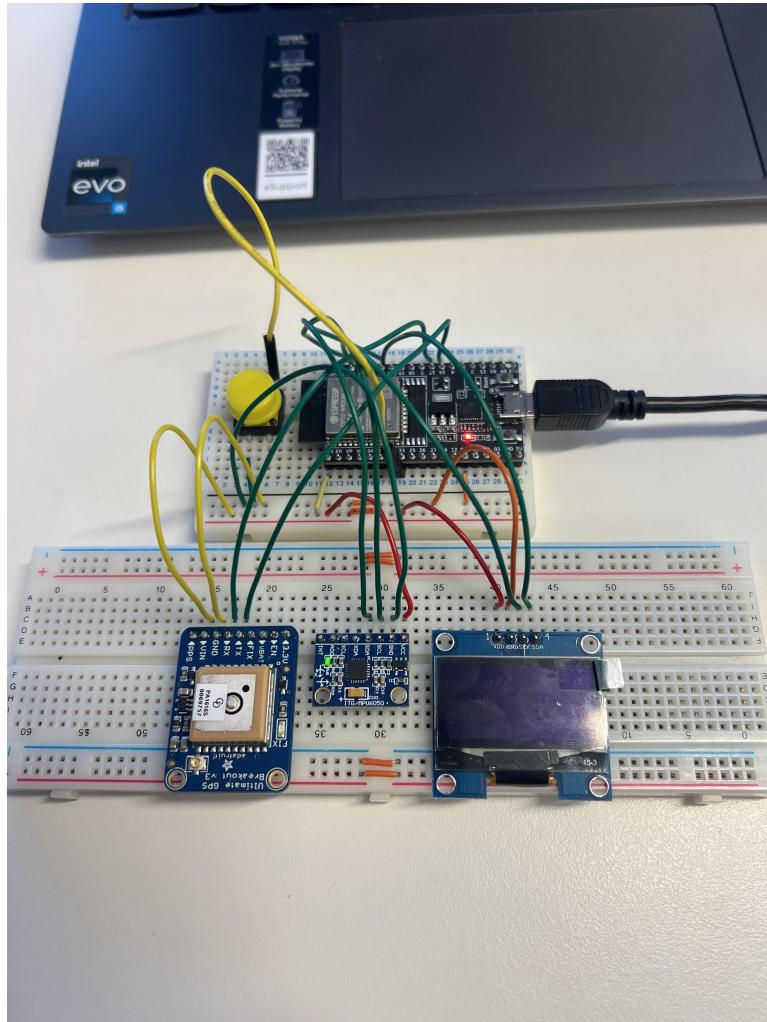
3 Test resultat og diskusjon

Resultatene fra koden ligger da under, dette er et bilde fra Ubidots nettsiden der vi har fått publisert dataen ESP32 innhenter, og viser dette på nettsiden:



Figur 3: Data på Ubidots

Her kan vi se at lokasjonen er printet til nettsiden, samtidig er det printet GPS-kvalitet, antall satelitter, akselerasjon, score og en akselerasjon over tid graf som også er nyttig informasjon for oss.



Figur 4: Bilde av oppkobling

4 Begrensninger og tiltak

Begrensninger til prototype 3:

- Kan ikke styre når vi sender dataen, sender hele tiden til vi trekker ut kabelen.
- Mangler en feedback på kjøringen.
- Dataen som sendes til Ubidots er ukryptert.
- Prosjektet funker ikke tilkobling til wifi.

Eventuelle tiltak:

- Implementere en knapp som styrer når vi sender data til Ubidots.
- Lage en ”score” funksjon som overvåker kjøringsmønsteret og gir oss feedback.
- Et tiltak er å kryperere dataen som blir sendt til Ubidots.

5 Konklusjon

Programmet funker som vi vil, det tar inn data fra både akselerometer og GPS-Tracker og poster dette til Ubidots på en oversiktlig måte.