ELE 402 - GRADUATION PROJECT II
INTERIM REPORT

HACETTEPE UNIVERSITY
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

PROJECT TITLE: Design of GNSS Aided Inertial Navigation System

PROJECT GROUP MEMBER: Barış Çapat

PROJECT SUPERVISOR: Dr. Yakup Özkazanç

SUBMISSION DATE: 12/12/2022

FALL 2022-2023

TABLE OF CONTENTS

1. INTRODUCTION

In the first part of the project, I made a theoretical research on how to approach the system I will design. The system consists of two main parts; IMU: the part that allows us to get the acceleration, yaw, pitch and roll angles, GPS: the part where we will calculate the speed together with the longtitude, altitude and heading. Along with the rest of the parts and sensors, I made the theoretical studies that we can run these two parts together and get the GNSS aided Inertial system simultaneously via the microcontroller.

As a result of these studies, I bought the necessary equipment, as IMU, I bought a DF-robot 10 DOF Mems sensor with stable values, which can give me more than the required data, it was a good option for me to see my mistakes in the applications I would make, since I could access certain examples with the Arduino I used. I bought the GY-NEO6MV2 GPS Module as a GPS module, it is a model that comes with an antenna and is budget friendly, in addition, it is an advantage for me to have certain application examples together with the arduino model I use.

In addition to these, I needed a power source to run the system and I did not need an external battery and circuit as the powerbank Xiaomi 10000 mAh model is ideal for mobility and has a circuit that does not turn itself off at small currents. I'm thinking of adding SD card module and wifi card for the later and final parts of the project, so I can actively transfer data to the computer and observe or write to the sd card.

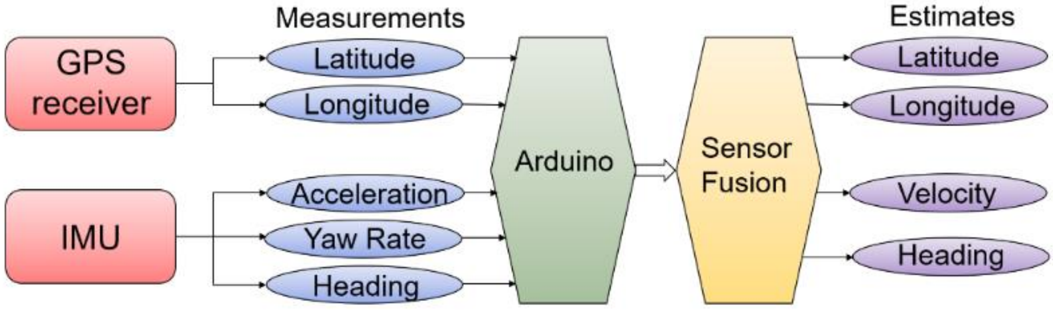


Figure 1: *Block diagram for GPS and IMU*

2. PROTOTYPE

The system I designed as the first prototype is actually a very simple design, IMU and GPS are connected to the arduino uno board via jumpers. When I manage to run the GPS and IMU simultaneously with the arduino in the last parts, I will connect a wifi card or sd card to this system and put it in a small portable box after the necessary soldering processes and make it modular.

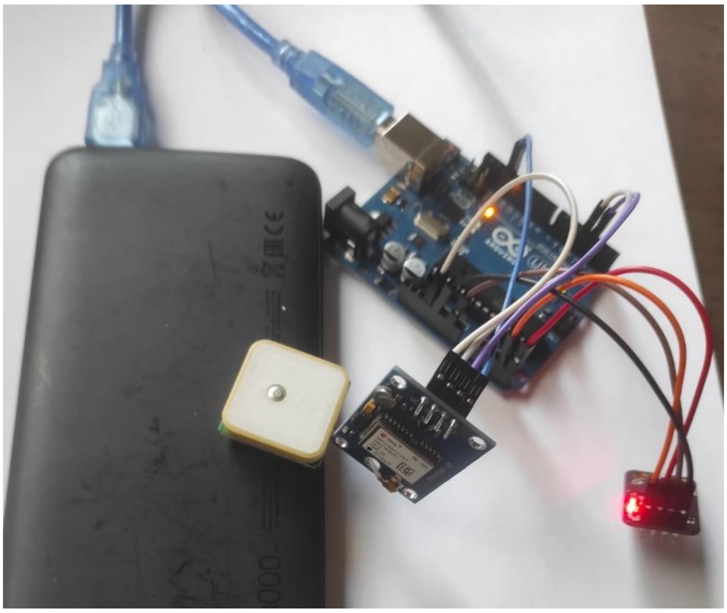


Figure 2: *Connection of the first prototype for GPS and IMU*

2.1.APPLICATION OF THE IMU

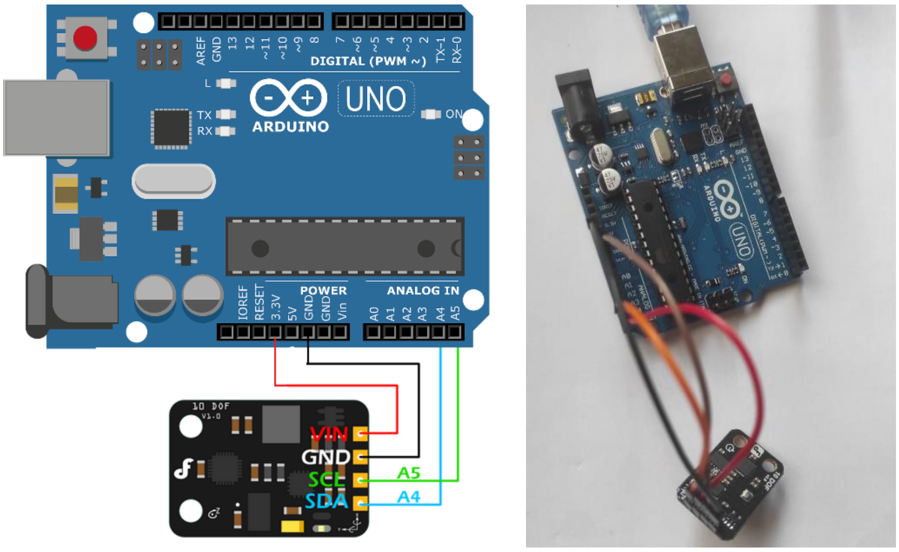


Figure 3: *Photo and diagram of the IMU connection*

2.1.1. TESTING AND RESULTS

3.1.1.1_getting x,y,z axises accelerations

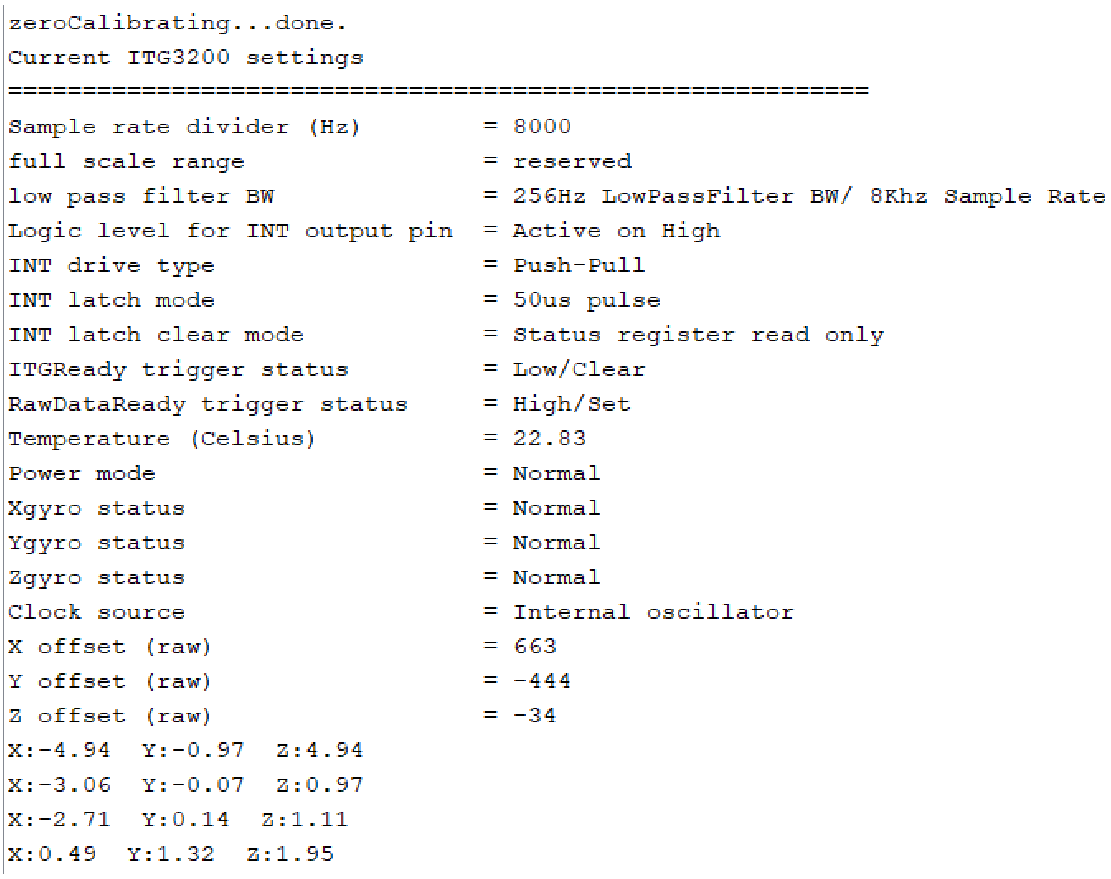
```
#include <DFRobot_ITG3200.h>
#define I2C_ADDR 0x68
DFRobot_ITG3200 gyro = DFRobot_ITG3200(&Wire, I2C_ADDR);
float xyz[3], temperature;
void setup(void)
{
  Serial.begin(9600);
  gyro.begin();
  gyro.reset();
  /**
   * @brief set gyro sampling rate
   * @n     Sampling rate = 1kHz / (7 + 1) = 125Hz
   */
  // gyro.setSamplerateDiv(/*Sample rate divider*/7);
  /**
   * @brief get gyro sampling rate
   */
  // gyro.getSamplerateDiv();
  /**
   * @brief set the interrupt pin level when an interrupt is triggered
   */
  // gyro.setIntlogicLvl(false);
  /**
   * @brief Check if the interrupt pin is at low level when an interrupt is triggered
   */
  // gyro.isIntactiveOnlow();
  /**
   * @brief set interrupt pin status
   * @n     OPEN_DRAIN
   * @n     PUSH_PULL
   */
  // gyro.setIntdriveType(OPEN_DRAIN);
  /**
   * @brief whether the interrupt pin is open drain output
   */
  // gyro.isIntopenDrain();

  Serial.print("zeroCalibrating...");
  gyro.zeroCalibrate(2500, 2);
  Serial.println("done.");
  showall();
  delay(5000);
}
void loop(void)
{
  while (gyro.isRawdataReady())
  {
    gyro.readGyro(xyz);
    Serial.print("X:");
    Serial.print(xyz[0]);
    Serial.print("  Y:");
    Serial.print(xyz[1]);
    Serial.print("  Z:");
    Serial.println(xyz[2]);
  }
}
void showall(void)
{
  Serial.println("Current ITG3200 settings");
  Serial.println("====================
====================================
==");
  Serial.print("Sample rate divider
(Hz)        = ");
  if (gyro.getFilterBW() ==
BW256_SR8)
    Serial.println(8000 /
(gyro.getSamplerateDiv() + 1), DEC);
  else
    Serial.println(1000 /
(gyro.getSamplerateDiv() + 1), DEC);
  Serial.print("full scale range
= ");
  if (gyro.getFSrange() ==
RANGE2000)
    Serial.println("+-2000
deg/sec");
  else
    Serial.println("reserved");
  Serial.print("low pass filter BW
= ");
  switch (gyro.getFilterBW())
  {
    case BW256_SR8:
      Serial.println("256Hz
LowPassFilter BW/ 8Khz Sample
Rate");
      break;
    case BW188_SR1:
      Serial.println("188Hz
LowPassFilter BW/ 1Khz Sample
Rate");
      break;
    case BW098_SR1:
      Serial.println("98Hz
LowPassFilter BW/ 1Khz Sample
Rate");
      break;
    case BW042_SR1:
      Serial.println("42Hz
LowPassFilter BW/ 1Khz Sample
Rate");
      break;
    case BW020_SR1:
      Serial.println("20Hz
LowPassFilter BW/ 1Khz Sample
Rate");
      break;
    case BW010_SR1:
      Serial.println("10Hz
LowPassFilter BW/ 1Khz Sample
Rate");
      break;
    case BW005_SR1:
      Serial.println("5Hz
LowPassFilter BW/ 1Khz Sample
Rate");
      break;
  }
  Serial.print("Logic level for INT
output pin  = ");
  if (gyro.isIntactiveOnlow())
    Serial.println("Active on Low");
  else
    Serial.println("Active on
High");
  Serial.print("INT drive type
= ");
  if (gyro.isIntopenDrain())
    Serial.println("Open Drain");
  else
    Serial.println("Push-Pull");

  Serial.print("INT latch mode
= ");
  if (gyro.isLatchuntilCleared())
    Serial.println("Latch until
interrupt is cleared");
  else
    Serial.println("50us pulse");
  Serial.print("INT latch clear mode
= ");
```

```
  if (gyro.isAnyregClrmode())
    Serial.println("Any register
read");
  else
    Serial.println("Status register
read only");
  Serial.print("ITGReady trigger
status          = ");
  if (gyro.isItgreadyOn())
    Serial.println("High/Set");
  else
    Serial.println("Low/Clear");
  Serial.print("RawDataReady trigger
status     = ");
  if (gyro.isRawdataReady())
    Serial.println("High/Set");
  else
    Serial.println("Low/Clear");
  Serial.print("Temperature
(Celsius)           = ");
  gyro.readTemp(&temperature);
  Serial.println(temperature);
  Serial.print("Power mode
= ");
  gyro.setPowermode(NORMAL);
  if (gyro.isLowpower() == STANDBY)
    Serial.println("Low power
(sleep)");
  else
    Serial.println("Normal");
  Serial.print("Xgyro status
= ");
  if (gyro.isXgyroStandby() ==
NORMAL)
    Serial.println("Normal");
  else
    Serial.println("StandBy");
  Serial.print("Ygyro status
= ");
  if (gyro.isYgyroStandby() ==
NORMAL)
    Serial.println("Normal");
  else
    Serial.println("StandBy");
  Serial.print("Zgyro status
= ");
  if (gyro.isZgyroStandby() ==
NORMAL)
    Serial.println("Normal");
  else
    Serial.println("StandBy");
  Serial.print("Clock source
= ");
  switch (gyro.getClocksource())
  {
    case INTERNALOSC:
      Serial.println("Internal
oscillator");
      break;
    case PLL_XGYRO_REF:
      Serial.println("PLL with X
Gyro reference");
      break;
    case PLL_YGYRO_REF:
      Serial.println("PLL with Y
Gyro reference");
      break;
    case PLL_ZGYRO_REF:
      Serial.println("PLL with Z
Gyro reference");
      break;
    case PLL_EXTERNAL32:
      Serial.println("PLL with
external 32.768kHz reference");
      break;
    case PLL_EXTERNAL19:
      Serial.println("PLL with
external 19.2MHz reference");
      break;
  }

  Serial.print("X offset (raw)
= ");

  Serial.println(gyro.offsets[0]);

  Serial.print("Y offset (raw)
= ");

  Serial.println(gyro.offsets[1]);

  Serial.print("Z offset (raw)
= ");

  Serial.println(gyro.offsets[2]);

}
```

3.1.1.2_Getting yaw, roll, pitch angles

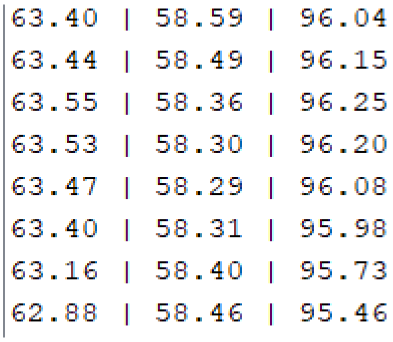
```
#include <FreeSixIMU.h>

#include <FIMU_ADXL345.h>

#include <FIMU_ITG3200.h>

#include <Wire.h>

float angles[3]; // yaw pitch roll

// Set the FreeSixIMU object

FreeSixIMU sixDOF = FreeSixIMU();

void setup() {

  Serial.begin(9600);

  Wire.begin();

  delay(5);

  sixDOF.init(); //begin the IMU

  delay(5);

}

void loop() {

  sixDOF.getEuler(angles);

  Serial.print(angles[0]);

  Serial.print(" | ");

  Serial.print(angles[1]);

  Serial.print(" | ");

  Serial.println(angles[2]);

  delay(100);}
```

2.1.2. EVALUATION



```
zeroCalibrating...done.
Current ITG3200 settings
==========================================================
Sample rate divider (Hz)         = 8000
full scale range                 = reserved
low pass filter BW               = 256Hz LowPassFilter BW/ 8Khz Sample Rate
Logic level for INT output pin   = Active on High
INT drive type                   = Push-Pull
INT latch mode                   = 50us pulse
INT latch clear mode             = Status register read only
ITGReady trigger status          = Low/Clear
RawDataReady trigger status      = High/Set
Temperature (Celsius)            = 22.83
Power mode                       = Normal
Xgyro status                     = Normal
Ygyro status                     = Normal
Zgyro status                     = Normal
Clock source                     = Internal oscillator
X offset (raw)                   = 663
Y offset (raw)                   = -444
Z offset (raw)                   = -34
X:-4.94  Y:-0.97  Z:4.94
X:-3.06  Y:-0.07  Z:0.97
X:-2.71  Y:0.14  Z:1.11
X:0.49  Y:1.32  Z:1.95
```

Figure 4: *3.1.1.1 x,y,z axises acceleration*



```
63.40 | 58.59 | 96.04
63.44 | 58.49 | 96.15
63.55 | 58.36 | 96.25
63.53 | 58.30 | 96.20
63.47 | 58.29 | 96.08
63.40 | 58.31 | 95.98
63.16 | 58.40 | 95.73
62.88 | 58.46 | 95.46
```

Figure 5: *3.1.1.2 roll, pitch, yaw angles*

2.2.APPLICATION OF THE GPS

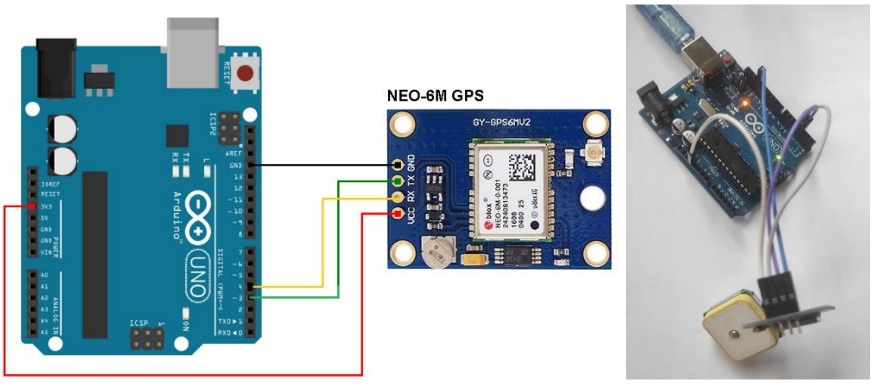


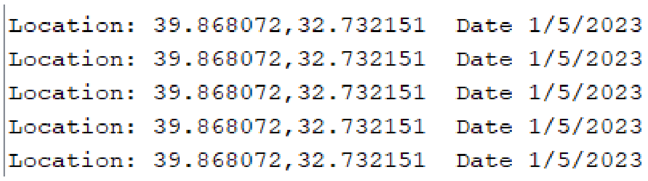
__Figure 6:__ Photo and diagram of the GPS connection

2.2.1. TESTING AND RESULTS

```
#include <TinyGPSPlus.h>

#include <SoftwareSerial.h>

static const int RXPin = 3, TXPin =
4;// Here we make pin 4 as RX of
arduino & pin 3 as TX of arduino

static const uint32_t GPSBaud = 9600;

TinyGPSPlus gps;

SoftwareSerial ss(RXPin, TXPin);

void setup()

{

  Serial.begin(9600);

  ss.begin(GPSBaud);

}

void loop()

{

  while (ss.available() > 0)

    if (gps.encode(ss.read()))

      displayInfo();

  if (millis() > 5000 &&
gps.charsProcessed() < 10)

  {

    Serial.println(F("No GPS
detected: check wiring."));

    while(true);}}
```

```
void displayInfo()

{

Serial.print(F("Location: "));

  if (gps.location.isValid())

  {

Serial.print(gps.location.lat(), 6);

    Serial.print(F(","));

Serial.print(gps.location.lng(), 6);

  }

   else

  {Serial.print(F("INVALID"));}

  Serial.print(F("  Date "));

  if (gps.date.isValid())

  {

    Serial.print(gps.date.month());

    Serial.print(F("/"));

    Serial.print(gps.date.day());

    Serial.print(F("/"));

    Serial.print(gps.date.year());

  }

  else

  {Serial.print(F("INVALID"));}

  Serial.println();}
```

2.2.2. EVALUATION



```
Location: 39.868072,32.732151  Date 1/5/2023
Location: 39.868072,32.732151  Date 1/5/2023
Location: 39.868072,32.732151  Date 1/5/2023
Location: 39.868072,32.732151  Date 1/5/2023
Location: 39.868072,32.732151  Date 1/5/2023
```

Figure 6: *3.2.1 altitude and longtitude values*

RESULTS

As a result, I can stably capture the data I want separately from GPS and IMU. After this stage, I hope to process the two data simultaneously and actively save the data I receive and see it on the screen. The parts where I will get errors or problems will be after this, but I still look for solutions. I'll put the latest system in a box that's portable and can run for as long as needed.

REFERENCES

https://www.youtube.com/watch?v=3Ng20FycQcY
https://wiki.dfrobot.com/10_DOF_Sensor__SKU_SEN0140_
https://www.youtube.com/watch?v=zsBmkO8wZ60
https://github.com/topics/gps?l=c%2B%2B
https://pdf.direnc.net/upload/10-dof-mems-imu-sensor-datasheet.pdf
https://github.com/DFRobot/FreeSixIMU
https://projecthub.arduino.cc/
https://www.adafruit.com/product/2472
https://www.tutorialspoint.com/interfacing-gnss-receiver-with-arduino-to-get-location
https://www.instructables.com/Building-a-GNSS-Receiver-Using-MosaicHAT-and-Ardui/
https://www.researchgate.net/publication/329371876_GPS_IMU_Data_Logger_with_Arduino_Nano
https://www.instructables.com/Complete-Arduino-based-Vehicle-GPSGPRS-Anti-theft-/
https://www.youtube.com/watch?v=DpYY1xlSEts
https://www.youtube.com/watch?v=lWJpM2tbh_U
https://www.scielo.org.mx/pdf/jart/v12n4/v12n4a17.pdf
https://www.dfrobot.com/blog-834.html
https://www.sparkfun.com/datasheets/Sensors/Gyro/PS-ITG-3200-00-01.4.pdf
https://www.youtube.com/watch?v=KMhbV1p3MWk
https://www.youtube.com/watch?v=lclVFN_ASWs
http://aprs.gids.nl/nmea/#gll
https://www.youtube.com/watch?v=pVcjXIG4KW8
https://swairlearn.bluecover.pt/nmea_analyser
https://www.latlong.net/Show-Latitude-Longitude.html
https://www.youtube.com/watch?v=DJLHo2kEJH0
https://www.davidjwatts.com/youtube/GPS-software-serial.ino
https://www.youtube.com/watch?v=bgOZLgaLa0g
https://www.esp8266.com/viewtopic.php?f=160&t=21488
https://www.youtube.com/watch?v=_eqdatIPprw