**GEÏNTEGREERDE PROEF**

BIKE-ESP

Sybe Van den Bergh

**Studierichting ●**

**Tweede leerjaar ● derde graad**

**Schooljaar 2025-2026**

**Klassenleraar: Bart Huyskens**

Inhoud

[1. Voorwoord 3](#_Toc212020595)

[2. Inleiding en bedanking 3](#_Toc212020596)

[3.1 Wetenschappelijke methode 3](#_Toc212020597)

[3.1.1 Kaartweergave 3](#_Toc212020598)

[3.1.2 Touchscreen TFT display 12](#_Toc212020599)

[3.1.3 GPS 14](#_Toc212020600)

[3.2 Engineering methode 16](#_Toc212020601)

[4. Bronnenlijst 17](#_Toc212020602)

[5. Conclusie & aanbevelingen 18](#_Toc212020603)

# Voorwoord

# Inleiding en bedanking

* De leerling omschrijft in 5 tot 10 zinnen wat hij/zij beoogt te verwezenlijken of reeds heeft gerealiseerd in de GIP.
* Dit wordt ondersteund door een visuele illustratie, zoals een blokschema, tekening of foto, om het proces of resultaat duidelijk weer te geven.
* De leerling bedankt kort de personen die hebben bijgedragen aan de realisatie van de GIP en vermeldt daarbij specifiek welke hulp of ondersteuning zij hebben geboden.

# 3.1 Wetenschappelijke methode

### 3.1.1 Kaartweergave

* **Onderzoeksvraag**
  + Hoe kan de kaart gedecodeerd worden van PNG naar een leesbaar bestand voor de ESP32?
  + Hoe kunnen de gedecodeerde bestanden gedraaid en aangevuld worden zodat de gebruiker altijd in het midden staat?
* **Informeren**
  + **OpenStreetMap**

OpenStreetMap is een organisatie die gratis kaarten ter beschikking stelt. Hiervoor dient men wel een vermelding te maken vanwaar ze komen. De kaart is beschikbaar in 2 vormen

* + - **Vectorkaarten:** Dit zijn vectorbestanden die omgezet moeten worden zodat de data makkelijk uit deze bestand kan genomen worden. Dit kan gebeuren op een gewone computer.
    - Afbeelding met tekst, kaart, diagram, atlas

      Door AI gegenereerde inhoud is mogelijk onjuist.**Tiles:** Dit zijn kleine png afbeeldingen van 256px op 256px. Ze kunnen voor veel toepassingen gebruikt worden.

Voor dit project wordt er gewerkt met ‘Tiles’. Deze kunnen gedownload worden met Maperative. Dit is een gratis software waarbij een specifiek gebied gekozen kan worden om dit dan de tiles te downloaden. Hiervoor moet het minimum zoomlevel en het maximum zoomlevel voor opgegeven worden. Het wordt dan geëxporteerd in volgende structuur.

📁 MAP

└── 📁 zoom\_level (nummer)

└── 📁 tile\_X (nummer)

└── 🗺️ tile\_Y.png

* **PNG decodering**

Om de geëxporteerde tiles te kunnen gebruiken moeten deze eerst gedecodeerd worden naar een formaat dat het TFT-display kan gebruiken. Hiervoor zijn twee manieren mogelijk.

* + **Decoderen via externe bibliotheek:** Voor deze manier is een aparte bibliotheek nodig, deze decodeert dan de png tot een RGB565 array.
  + **Decoderen via display driver:** Er zijn niet veel display drivers die png’s kunnen decoderen. Op dit moment is er mij maar 1 bekend die dit kan. Deze driver LovyanGFX werkt met een Sprite systeem. Een Sprite wordt aangemaakt en daarin kan de png geladen worden. Hiervoor heeft de bibliotheek een interne decoder.

Hieronder is een tabel beschikbaar met testresultaten op elke manier:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 tile (256x256) | **Pngle** | **PNGDec** | **LovyanGFX** |
| Test 1 |  |  |  |
| Test 2 |  |  |  |
| Test 3 |  |  |  |
| Test 4 |  |  |  |
| Test 5 |  |  |  |

* **Positionering op de kaart**

Als de png’s gedecodeerd zijn moeten de tiles nog samengevoegd worden zodat de gebruiker altijd in het midden van de kaart staat. Hier kan ook nog aan toegevoegd worden om de kaart te draaien. Hierdoor kan de gebruiker makkelijker navigeren omdat wat hij ziet overeenkomt met de werkelijkheid.

Om dit te realiseren zijn een paar stappen nodig.

* + **Positie bepalen:** Als eerste stap moet de positie van de gebruiker opgehaald worden. Dit gebeurt met de GPS.
  + **Tiles ophalen:** Als de positie is opgehaald kunnen de juiste tiles opgehaald worden. De tiles die opgehaald moeten worden is zeker de hoofdtile, dit is de tile waar de gebruiker zich op bevind. Maar ook de rond omliggende tiles moeten opgehaald worden. Voor het bepalen van de hoofdtile kan volgende berekening uitgevoerd worden.
* n = 2 ^ zoom
* xtile = n \* ((lon\_deg + 180) / 360)
* ytile = n \* (1 - (log(tan(lat\_rad) + sec(lat\_rad)) / π)) / 2
  + **Offset berekenen:** In deze stap wordt bereken waar de gebruiker juist staat in de hoofdtile. Dit wordt gedaan met de getallen na de komma, dit in de vorige berekening gebeurt zijn. Dit geeft het aantal procent dat de gebruiker afwijkt. Wat nadien omgezet kan worden naar de juiste pixel.
  + **Tiles samenvoegen:** Voor het samenvoegen van de tiles wordt een Sprite gemaakt die groter is dan de oppervlakte van het scherm, hier worden de tiles op de juiste positie geplaatst. De kaart wordt ook gedraaid indien nodig. Nadien wordt het juiste deel van de Sprite op het scherm gezet.Afbeelding met diagram, lijn, schets, Plan

    Door AI gegenereerde inhoud is mogelijk onjuist.
* **Experimenteren**

Alle code die hierachter komt is code die nog niet getest is. Dit omdat deze code gemaakt is voor een ESP32. Deze zijn krachtiger en kunnen de png wel decoderen.

#### **Main.cpp**

#include <Arduino.h>

#include "tft.hpp"

#include "mapSimple.hpp"

#include "mapFull.hpp"

#include <SPI.h>

#include <SD.h>

TFT\_eSprite img = TFT\_eSprite(&tft);

mapSimple mapS;

mapFull mapF;

void setup() {

// put your setup code here, to run once:

initTFT();

if (!SD.begin(GPIO\_NUM\_5)) {

  Serial.println("Card failed, or not present");

  while (1);

}

}

void loop() {

  // put your main code here, to run repeatedly:

  bool simpelmap = true;

  double lon = 51.2478;

  double lat = 4.494555;

  int z = 15;

  if(simpelmap){

    mapS.set1TileInBuffer(z, lon, lat, img);

  } else {

    mapF.makeMap(z, lon, lat, img);

  }

}

**Tft.cpp**

#include "tft.hpp"

TFT\_eSPI tft = TFT\_eSPI();

uint16\_t TFT\_WIDTH = 0;

uint16\_t TFT\_HEIGHT = 0;

void initTFT()

{

    tft.init();

    TFT\_HEIGHT = tft.height();

    TFT\_WIDTH = tft.width();

    tft.initDMA();

    tft.fillScreen(TFT\_BLACK);

}

**Tft.hpp**

#pragma once

#include <Arduino.h>

#include "display.hpp"

#include <LGFX\_TFT\_eSPI.hpp>

extern TFT\_eSPI tft;

extern uint16\_t TFT\_WIDTH;

extern uint16\_t TFT\_HEIGHT

void initTFT();

**Display.hpp**

#pragma once

#define LGFX\_USE\_V1

#include <LovyanGFX.hpp>

class LGFX : public lgfx::LGFX\_Device

{

  lgfx::Panel\_ILI9488   \_panel\_instance;

  lgfx::Light\_PWM       \_light\_instance;

  lgfx::Touch\_FT5x06    \_touch\_instance;

  public:

  LGFX(void)

  {

    {

      auto cfg = \_panel\_instance.config();

      cfg.pin\_cs = -1;

      cfg.pin\_rst = -1;

      cfg.pin\_busy = -1;

      cfg.memory\_width = 320;

      cfg.memory\_height = 480;

      cfg.panel\_width = 320;

      cfg.panel\_height = 480;

      cfg.offset\_x = 0;

      cfg.offset\_y = 0;

      cfg.offset\_rotation = 0;

      cfg.dummy\_read\_pixel = 8;

      cfg.dummy\_read\_bits = 1;

      cfg.readable = true;

      cfg.invert = true;

      cfg.rgb\_order = false;

      cfg.dlen\_16bit = true;

      cfg.bus\_shared = true;

      \_panel\_instance.config(cfg);

    }

    {

      auto cfg = \_light\_instance.config();

      cfg.pin\_bl = GPIO\_NUM\_10;

      cfg.invert = false;

      cfg.freq = 44100;

      cfg.pwm\_channel = 7;

      \_light\_instance.config(cfg);

      \_panel\_instance.setLight(&\_light\_instance);

    }

    {

      auto cfg = \_touch\_instance.config();

      cfg.x\_min = 0;

      cfg.x\_max = 319;

      cfg.y\_min = 0;

      cfg.y\_max = 479;

      cfg.pin\_int = GPIO\_NUM\_30;

      cfg.bus\_shared = true;

      cfg.offset\_rotation = 0;

      cfg.i2c\_port = 0;

      cfg.i2c\_addr = 0x38;

      cfg.pin\_sda = GPIO\_NUM\_38;

      cfg.pin\_scl = GPIO\_NUM\_39;

      cfg.freq = 400000;

      \_touch\_instance.config(cfg);

      \_panel\_instance.setTouch(&\_touch\_instance);

    }

    setPanel(&\_panel\_instance);

  }

};

**mapSimple.cpp**

#include "mapSimple.hpp"

int mapSimple::long2tilex(double lon, int z)

{

    return (int)(floor((lon + 180.0) / 360.0 \* (1 << z)));

}

int mapSimple::lat2tiley(double lat, int z)

{

    double latrad = lat \* M\_PI/180.0;

    return (int)(floor((1.0 - asinh(tan(latrad)) / M\_PI) / 2.0 \* (1 << z)));

}

void mapSimple::getTileNumber(double lon, double lat, int z, int &x, int &y)

{

    x = long2tilex(lon, z);

    y = lat2tiley(lat, z);

}

void mapSimple::set1TileInBuffer(int zoom, double lon, double lat, TFT\_eSprite &img)

{

    img.createSprite(256, 256);

    getTileNumber(lon, lat, zoom, x, y);

    String path = "/MAP/" + String(zoom) + "/" + String(x) + "/" + String(y) + ".png";

    img.drawPngFile(path.c\_str() , 0, 0);

}

**mapSimple.hpp**

#pragma once

#include <Arduino.h>

#include "display.hpp"

#include <LGFX\_TFT\_eSPI.hpp>

class mapSimple {

    public:

        void set1TileInBuffer(int zoom, double lon, double lat, TFT\_eSprite &img);

    private:

        int x, y;

        void getTileNumber(double lon, double lat, int z, int &x, int &y);

        int long2tilex(double lon, int z);

        int lat2tiley(double lat, int z);

};

**mapFull.cpp**

#include "mapFull.hpp"

int mapFull::long2tilex(double lon, int z) {

    return (int)(floor((lon + 180.0) / 360.0 \* (1 << z)));

}

int mapFull::lat2tiley(double lat, int z) {

    double latrad = lat \* M\_PI/180.0;

    return (int)(floor((1.0 - asinh(tan(latrad)) / M\_PI) / 2.0 \* (1 << z)));

}

void mapFull::makeMap(int z, double lon, double lat, TFT\_eSprite &img) {

    img.createSprite(bufferSizeX, bufferSizeY);

    int centerTileX = long2tilex(lon, z);

    int centerTileY = lat2tiley(lat, z);

    int startX = centerTileX - tilesX / 2;

    int startY = centerTileY - tilesY / 2;

    img.fillSprite(TFT\_BLACK);

    for (int x = 0; x < tilesX; x++) {

        for (int y = 0; y < tilesY; y++) {

            int tileX = startX + x;

            int tileY = startY + y;

            String path = "/MAP/" + String(z) + "/" + String(tileX) + "/" + String(tileY) + ".png";

            int posX = x \* tileSize;

            int posY = y \* tileSize;

            img.drawPngFile(path.c\_str(), posX, posY);

        }

    }

}

**mapFull.hpp**

#pragma once

#include <Arduino.h>

#include "tft.hpp"

class mapFull {

public:

    mapFull() {}

    void makeMap(int z, double lon, double lat, TFT\_eSprite &img);

private:

    const int tileSize = 256;

    const int screenSizeX = 320;

    const int screenSizeY = 450;

    int bufferSizeX = 544;

    int bufferSizeY = 544;

    int tilesX = (bufferSizeX / tileSize) + 2;

    int tilesY = (bufferSizeY / tileSize) + 2;

    int long2tilex(double lon, int z);

    int lat2tiley(double lat, int z);

};

* **Reflecteren**
* **Rapporteren**
  + De leerling vat beknopt de opgedane kennis uit het onderzoek samen, waarbij de belangrijkste inzichten en resultaten worden toegelicht

### 3.1.2 Touchscreen TFT display

* **Onderzoeksvraag**
  + Hoe werkt de touchscreen van een LCD display?
* **Informeren**
  + Door gebruik te maken van een touchscreen kan het makkelijker zijn om het product te gebruiken. Een touchscreen kan op verschillende manieren werken. De twee bekendste zijn resistieve touchscreens en capacitieve touchscreens
  + **Resistieve touchscreens:**

Bij een resistieve touchscreen werkt het scherm op basis van stroom geleidende panelen. Tussen deze panelen zitten een soort isolatie bolletjes die ervoor zorgen dat ze elkaar niet kunnen aanraken. Als er op het scherm gedrukt wordt raken de panelen elkaar aan.

Afbeelding met tekst, schermopname, diagram, lijn

Door AI gegenereerde inhoud is mogelijk onjuist.

De panelen bestaan uit veel kleine draadjes die elk een lineaire weerstand hebben. Als er gedrukt wordt en er raken twee draadjes elkaar veranderen de elektrische eigenschappen van de draadjes en kan er een x en y waarde bepaald worden.

Afbeelding met schermopname, Rechthoek, diagram, tekst

Door AI gegenereerde inhoud is mogelijk onjuist.

* + **Capacitieve touchscreens:**

Een capacitief touchscreen werkt op basis van lading. Er zit bij dit soort schermen een toplaag over waar een constante lading op zit. De lading wordt op het scherm gezet vanuit de vier hoeken. Wanneer er op het scherm gedrukt wordt gaat er een kleine stroom lopen vanuit de vier hoeken. Het leest dan de waarde van de stroom en kan aan de hand van de stroomsterkte bereken waar op het scherm er iets aangeduid word.

Afbeelding met lijn, schermopname, cirkel, diagram

Door AI gegenereerde inhoud is mogelijk onjuist.

* **Experimenteren**
  + Praktisch experiment uitvoeren
  + Rapporteren
    - Aansluitschema – tekeningen – in hoge resolutie (vermijd screenshots)
    - Code in kleur – met commentaar – je kan elke regel 100% uitleggen
    - Metingen in goede resolutie – op witte achtergrond – met uitleg
* **Reflecteren**
  + Geven de resultaten van het experiment een antwoord op de onderzoeksvraag
* **Rapporteren**
  + De leerling vat beknopt de opgedane kennis uit het onderzoek samen, waarbij de belangrijkste inzichten en resultaten worden toegelicht

### 3.1.3 GPS

* **Onderzoeksvraag**
  + Hoe werkt het principe van een GPS, en hoe kan deze worden uitgelezen?
* **Informeren**
  + **GPS**

Gps is een digitaal systeem dat de positie op aarde kan bepalen door middel van satellieten. Deze zitten in een middelhoge baan rond de aarde. Ze verzenden hun positie, de tijd en een kloksignaal door naar de aarde. Vanaf een ontvanger drie satellieten in zicht heeft kan deze een positie vaststellen. De voorkeur gaat toch eender naar vier satellieten omdat het kloksignaal niet altijd even accuraat is.

Afbeelding met cirkel, kaart, diagram

Door AI gegenereerde inhoud is mogelijk onjuist.

* **Plaatsbepaling**

De plaatsbepaling gebeurt met behulp van driehoeksmeetkunde op een bol. Hiervoor heeft het de volgende data nodig: de positie van de satelliet op de moment van het verzenden, en de moment dat de data verzonden is. Hierdoor kan het bereken hoelang de data onderweg was en dus ook wat de afwijking is.

* **NMEA**

NMEA is het meest gebruikte protocol om GPS data te verzenden tussen de ontvanger en het systeem. De GPS chip zet de data die het ontvangt van een satelliet direct om naar NMEA code.

De data wordt dan via UART naar het systeem gestuurd. Hierbij is een standaard start van een bericht voorzien.

$[ SOORT positie systeem ]+[ Type data ]

* Soort positie system

|  |  |
| --- | --- |
| BD of GB | Beidou (Chineese versie) |
| GA | Galileo (Europese versie) |
| GP | GPS (Amerikaanse versie) |
| GL | Glonass (Russische versie) |

* Type data

|  |  |
| --- | --- |
| GGA | Tijd, positie en connectie gerelateerde data |
| GLL | Positie data |
| GSA | Active satellieten |
| GSV | Aantal zichtbare satellieten |
| RMC | Positie, snelheid en tijd |
| VTG | Richting en grondsnelheid |

In de inhoud van de datatypes zit veel overlap. Hierdoor kan je in verschillende berichten de zelfde data terug vinden. Er kan dus veel data uitgelezen worden met beperkt aantal berichten.

* Voorbeeld bericht
* $GPGGA,092750.000,5321.6802,N,00630.3372,W,1,8,1.03,61.7,M,55.2,M,,\*76
* **Experimenteren**

#include <TinyGPSPlus.h>

#include <SoftwareSerial.h>

static const int RXPin = 4, RXPin = 3;

static const uint32\_t GPSBaud = 4800;

TinyGPSPlus gps;

SoftwareSerial serialS(RXPin, TXPin); // Gebruik geen TX en RX pin, emulate TX en RX via software

void setup() {

  // put your setup code here, to run once:

  Serial.begin(9600);

  ss.begin(GPSBaud);

}

void loop() {

  // put your main code here, to run repeatedly:

  int satellieten = gps.satellites.value();

  float latitude = gps.location.lat();

  float longitude = gps.location.lng();

  float precision = gps.hdop.hdop(); // precisie van de locatie (lon, lat)

  //minder dan 2 is goed, hoger is slechte ontvangst

  float altitude = gps.altitude.meters();

  float heading = gps.course.deg();

  float speed = gps.speed.kmph();

}

* **Reflecteren**
  + Geven de resultaten van het experiment een antwoord op de onderzoeksvraag
* **Rapporteren**
  + De leerling vat beknopt de opgedane kennis uit het onderzoek samen, waarbij de belangrijkste inzichten en resultaten worden toegelicht

# Engineering methode

In deze fase beschrijft de leerling duidelijk wat hij of zij wil verwezenlijken. Hierbij worden de onzekerheden binnen het project benoemd, evenals de onderdelen die in het eerste ontwerp gerealiseerd zullen worden. Daarnaast geeft de leerling aan welke aspecten in toekomstige versies verder ontwikkeld en geoptimaliseerd zullen worden. Hierbij wordt onderstaande methode gebruikt voor drie ontwerpfasen:

#### Ontwerpfase 1 – 2 – 3

* **Probleemstelling (deze blijft de drie fasen gelijk)**
  + De leerling omschrijft duidelijk wat hij/zij met het project wil verwezenlijken.
  + De leerling benoemt expliciet de onderdelen waar nog onzekerheden over bestaan.
  + De leerling geeft aan welke onderdelen in het eerste ontwerp worden gerealiseerd en welke aspecten in toekomstige versies verder uitgewerkt zullen worden.
* **Ontwerpen** 
  + Versie 1, 2 of 3 van het ontwerp (inclusief tekeningen, berekeningen, schema’s, code, onderdelenlijst, bestelcodes en prijzen) wordt hier duidelijk weergegeven.
* **Realiseren** 
  + De leerling toont met foto’s en metingen aan wat hij/zij heeft gerealiseerd.
* **Uittesten** 
  + De leerling vermeldt hier duidelijk de testresultaten en geeft aan of de realisatie voldoet aan de vooraf gestelde verwachtingen.
* **Bijsturen** 
  + De leerling identificeert wat nog niet naar behoren functioneert (zoals vermeld in de tussentijdse procesevaluatie) en geeft aan welke aanpassingen hij/zij wil doorvoeren. Deze bijsturingen moeten zichtbaar zijn in de volgende fase van het ontwerp en de realisatie.

# Bronnenlijst

* OpenStreetMap
  + <https://wiki.openstreetmap.org/wiki/Tiles>
  + <https://github.com/kikuchan/pngle>
  + <https://github.com/bitbank2/PNGdec>
  + <https://github.com/lovyan03/LovyanGFX>
  + <https://wiki.openstreetmap.org/wiki/Slippy_map_tilenames>
* Touchscreen
  + <https://www.beamers-en-touchscreens.nl/news/wat-is-een-touchscreen/>
  + <https://www.sciencespace.nl/technologie/artikelen/5016/hoe-werkt-een-touchscreen>
* GPS
  + <https://landing.advancednavigation.com/tech-articles/global-navigation-satellite-system-gnss-and-satellite-navigation-explained/>
  + <https://en.wikipedia.org/wiki/NMEA_0183>
  + <https://www.nmea.org/nmea-0183.html>
  + <https://receiverhelp.trimble.com/alloy-gnss/en-us/NMEA-0183messages_MessageOverview.html>

# Conclusie & aanbevelingen

De leerling geeft een beknopte reflectie over de leerervaringen die zijn opgedaan tijdens de uitvoering van de GIP. Hierbij wordt aandacht besteed aan de volgende punten:

* De leerling beschrijft welke problemen zich tijdens het proces hebben voorgedaan en op welke wijze deze zijn opgelost.
* De leerling reflecteert op de aanpak en geeft aan wat hij/zij in een toekomstig project anders zou aanpakken.
* De leerling evalueert in hoeverre het uitdagend was om de planning te volgen en waar eventuele moeilijkheden optraden.

Elk van deze punten wordt ondersteund met visuele illustraties, zoals foto’s of beelden van het eindresultaat of de verschillende fasen van het project, om de uitleg te verduidelijken en te versterken.