

**EDUGO Campus Glorieux**

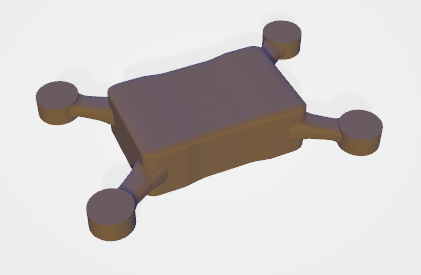
SCHOOL VOOR WETENSCHAP & TECHNIEK

Sint-Jozefstraat 7 9041 Gent-Oostakker

Tel. school : 09/255 91 15

**Verslag geïntegreerde proef**

**Drone**



Naam: *Rune Eerdekens*

Studierichting: *elektronica elektriciteit*

Klas: *6EE*

GIP-begeleiders: *dhr. Eggermond, dhr. Vallozie, dhr. Van Eetvelde*

Schooljaar: *2022-2023*

# Woord vooraf

Dit verslag is opgesteld als deel van onze geïntegreerde proef in het laatste jaar van het middelbaar waarin ik uitleg zal geven over mijn eindejaar project.

Mijn doel was om een GIP te maken waarmee ik mezelf zou kunnen uitdagen om een goed beeld te krijgen van wat ik allemaal kon op het einde van mijn 6de jaar. Daarom heb ik geprobeerd om zo weinig mogelijk het internet te gebruiken, dus veel wan wat u zal lezen zijn mijn eigen conclusies.

Ik wens volgende personen te bedanken voor alle hulp die zij me geboden hebben:

* Het directieteam;
* dhr. Eggermond, dhr. Van Eetvelde, dhr. Vallozie en dhr. Wulleman;
* Mijn vrienden en familie;
* Medeleerlingen;
* Mijn leerkrachten tot nu;

Tekstverwerkingsprogramma: Microsoft Office Word 2022

Printer: HP DeskJet 3762 All-in-One

# Inhoudsopgave

# Inleiding

Mijn naam is Rune Eerdekens ik zit in het 6de jaar Elektronica Elektriciteit en als laatstejaarsstudent kreeg ik samen met mijn klasgenoten de opdracht om een eindwerk te maken en verdedigen.

Het doel van deze proef is om een ontwerp naar jouw keuze te ontwerpen en realiseren dat binnen het vakgebied EE valt, voor mijn project koos ik voor het bouwen van een drone. Ik wist dat dit een grote opdracht ging zijn en heb veel plezier gehad met het bouwen van dit project.

In de loop van het uitdenken en bouwen/programmeren van mijn GIP ben ik verschillende ingewikkelde obstakels tegengekomen en overwonnen. Het was zeker een uitdaging wat het uiteindelijk leuk maakte voor mij.

In deze versie van mijn GIP kan nog veel verbeterd of toegevoegd worden maar daaraan zal ik een uniek hoofdstuk voor voorzien.

Uiteindelijk ben ik blij hoe het is geëindigd ook al had ik liever iets meer tijd gehad om aan uitbreidingen te werken zodat ik die ook zou kunnen presenteren in mijn verdediging.

# Praktijk

## Algemeen

Zoals eerder vermeld is mijn GIP een drone in dit hoofdstuk zal ik toelichten hoe hij in elkaar zit in grote lijnen.

## Kern

Zoals eerder vermeld wordt deze drone gestuurd door een zelfontworpen breadboard met 3 componenten. Een Gyroscoop/acceleratiemeter, Arduino Nano, en een HC-05 bluetooth ontvanger

### HC-05 bluetooth antenne

\*foto antenne

Dit zijn de oren van de drone hier ziet u een foto van hoe die eruitziet.

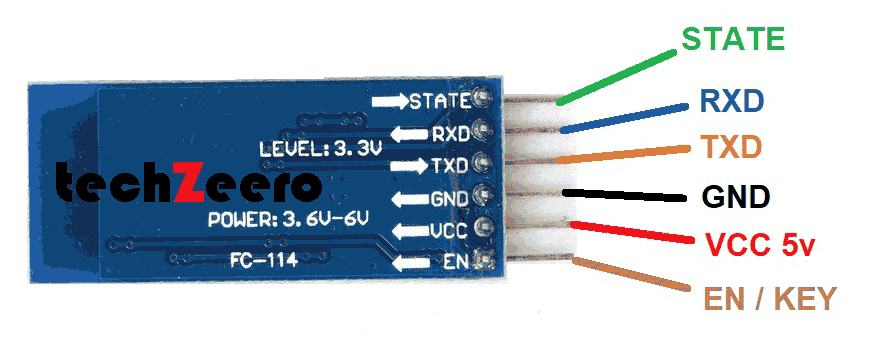


Foto 1: HC-05 Bluetooth module (www.theengineeringprojects.com)

Zoals u ziet op dit schema zijn vier van de zes pinnen verbonden deze zijn de enige die we nodig hebben voor dit project. De vier verbonden pinnen zijn de Vcc, de GND, de TX en de RX pinnen hun functie is als volgt:

* **Vcc**: +3.3 tot +6 Volt komt hiertoe, dit voed de component. In mijn project gebruik ik hiervoor de +5V van de Arduino.
* **GND**: dit is de ground pin deze gaat ook naar de ground van de Arduino
* **TX**: de TX pin is de 1ste van twee pinnen om data te communiceren naar de Arduino, deze word aan de RX pin van de Arduino gekoppeld.
* **RX**: net zoals de TX pin is deze nodig voor het praten met de Arduino deze word dus aan de TX pin gekoppeld.

De HC-05 module heeft nog twee andere pinnen die ik niet gebruik met volgende functies:

* **EN**: de enable pin van de module is actief laag dus als je een positieve spanning op deze pin zet zal de module dus de bluetooth afgezet worden.
* **State**: de state pin is heel handig aangezien deze hoog komt de staan als er een connectie is over bluetooth. Je kan deze gebruiken om in je Arduino code alleen maar dingen aan te zetten als bluetooth verbonden is.

In dit project gebruik ik bluetooth om te communiceren met de drone vanuit een app die op mijn GSM staat. Ook al heeft de HC-05 module de mogelijkheid om data te ontvangen en uit te sturen gebruik ik het alleen maar voor het ontvangen van data aangezien ik geen data terug stuur. Je zou de stuur functie bijvoorbeeld kunnen gebruiken om door te geven aan de gsm hoeveel procent de batterij nog heeft voordat hij plat is.

Er staat een voorbeeld hoe ik de module gebruik in mijn code op pagina X waar.

### MPU 6050 gyroscoop

\*foto gyroscoop

Dit is een foto van de gyroscoop deze component is verantwoordelijk voor het doorgeven van hoeks verandering en versnelling van de drone in alle drie de dimensies.

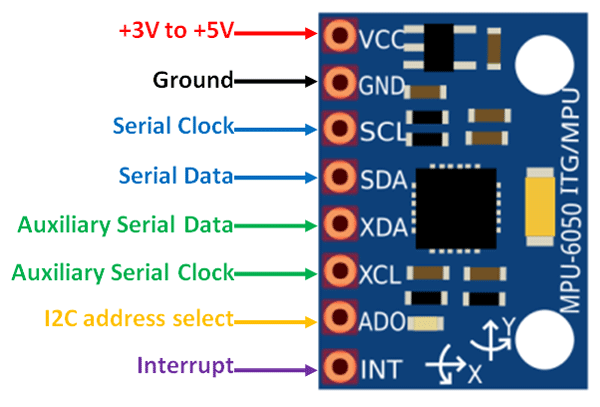


Foto 2: MPU-6050 gyroscoop (components101.com)

De MPU 6050 of gyroscoop heeft acht pinnen in totaal. Van deze acht gebruik ik er maar vier.

De vier pinnen die ik gebruik zijn de volgende:

* **Vcc**: zoals bij de HC-05 word deze gevoed door de Arduino en kan een spanning van +3V tot +5V
* **Ground**: ook zoals bij de HC-05 word deze pin aan de ground van de Arduino gekopeld.
* **SCL**: de seriële klok stuurt een kloksignaal door naar de Arduino zodat de Arduino de data kan inlezen bij de juiste frequentie en word aan de SCL van de Arduino gekoppeld
* **SDA**: de seriële data net zoals de klok is nodig voor het doorsturen van data van de MPU-6050 naar de Arduino en word aan de SDA pin van de Arduino gekoppeld.

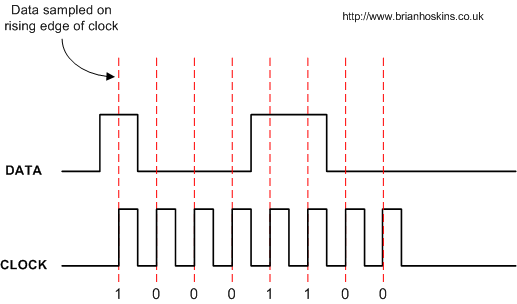


Foto 3: SDA en SCL voor data transfer (electronics.stackexchange.com)

In de foto ziet u dat de data alleen maar word ingelezen op de positieve flank van de klok. Zo wordt er verzekerd dat alles goed verloopt en dat je het juiste aantal bits leest. Als er bijvoorbeeld op de datalijn vier nullen na elkaar worden doorgestuurd kijk je bij elke positieve flank van de klok naar of de datalijn hoog (1) of laag (0) staat en zo kom je te weten hoeveel bits er in die lange lijn van nullen of één’ s zitten.

### Arduino Nano.

Dit is het brijn van de drone, alle wiskundige operaties worden op de Arduino uitgevoerd. Voor dit project gebruik ik een Arduino Nano Every, een variant niet veel verschillend van de normale Arduino Nano.

Afbeelding met tekst, schermopname, ontwerp

Automatisch gegenereerde beschrijving

Foto 4: Arduino Nano Every pin lay-out (wiki.rocrail.net)

Er zijn veel pinnen die niet worden gebruikt in dit project maar er zijn een paar belangrijk. Natuurlijk zijn de +5V en de GND (ground) belangrijk voor het voeden van alle componenten, en ook de TX / RX en SDA / SCL pinnen voor het doorsturen van data. Aangezien de motors van de drone BLDC (brushless DC) motoren zijn hebben we ook nog de PWM (puls width modulation) pinnen nodig om een signaal naar de motors te sturen die de motors begrijpen. Deze zijn aangeduid door een ~ teken naast het pin nummer. Degene die ik gebruik zijn D3, D5, D6 en D9. Er zijn vier PWM pinnen nodig omdat ik vier motors heb waarvan ik de snelheid individueel wil kunnen besturen. Het uploaden van code naar de Arduino wordt gedaan via een micro-USB. De TX en RX pinnen mogen wel niet verbonden zijn of de upload van de code faalt.

### Electronic Speed Controller

De volgende component is de Electronic Speed Controller of ESC in het kort. Hiervan heb ik er vier aangezien ik vier motoren heb.

Afbeelding met tekst, schermopname, Lettertype, lijn

Automatisch gegenereerde beschrijving

Foto : pin layout ESC

Zoals je ziet op het schema wordt de ESC gevoed met 7.4V vaan de 2S LiPo batterij. Dit is een soort batterij die vaak in drones wordt gebruikt voor hun energiedichtheid, ze zijn ook her oplaadbaar wat ze een perfecte kandidaat maken voor een Drone. Er zit op deze soort ESC een power regulator die handig is voor een regulated +5V te genereren om de Arduino met +5V te voeden. Dit is ook de component die het PWM signaal ontvangend van de Arduino en het omzet naar de 3 fase van de BLDC motor.

#### Werking PWM

Ik praat hier al een tijdje over PWM maar heb het nog niet duidelijk uitgelecht wat ik hier zal doen.

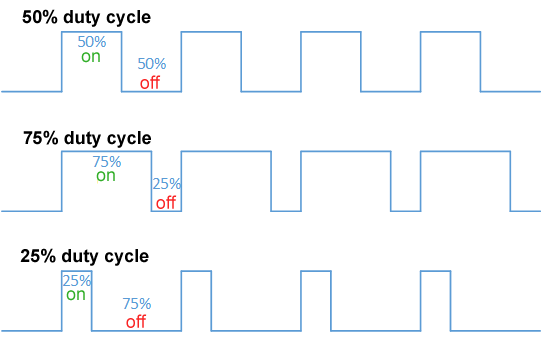


Foto : PWM signaal (en.wikipedia.org)

Een PWM signaal is opgebouwd uit pulsen. Deze pulsen zijn altijd een fractie van de duty cycle, als je in je PWM signaal doorstuurt waarbij de puls 100% is van de duty cycle zal de motor op volledige snelheid draaien, bij mijn ESC ligt de minimum puls grote op 700ms en da maximum op 2300ms dus als ik pulsen van 2300ms stuur zullen de motors op 100% snelheid draaien. Voor dit soort signaal te maken gebruik ik in mijn Arduino een bibliotheek die veel hiervoor wordt gebruikt genaamd <Servo.h>. Met deze bibliotheek kan ik de min en max puls grote instellen en dan later via een schaal van 0 tot 180 de grote van mijn pulsen aanpassen. Dus 0 zou een puls grote van 700ms zijn en 180 een grote van 2300ms.