

P&O Computerwetenschappen: Schrijfoopdracht 2

Team Indigo

Nu de opdracht en planning van dit project duidelijk is, zullen we in detail treden over de hardware- en softwarecomponenten.

Hardware

Alle onderdelen van de zeppelin zullen gemonteerd worden op een frame. Hierop worden 3 propellers bevestigd. Hiervan worden er twee gebruikt om naar links en rechts te draaien. Om vooruit te bewegen worden deze samen geactiveerd met dezelfde kracht. De derde propellor dient om de zeppelin te laten stijgen.

Om het geheel in de lucht te houden, worden er 2 ballonnen gebruikt. Deze hebben een diameter van ongeveer 90 centimeter en bevatten helium.

De zeppelin wordt aangestuurd door een Raspberry Pi model A. (zie Figuur 1) Deze heeft volgende specificaties:

- *Processor:* 700MHz ARM
- *Geheugen:* 256MB
- *Poorten:* 1 USB 2.0, HDMI, audio out, RCA video
- *Voeding:* Micro USB
- GPIO-pinnen om de hardware aan te sturen

In de Raspberry Pi zit een SD-kaart van 4 GB. Op de USB-poort is een USB hub aangesloten, zodat het mogelijk is meerdere onderdelen aan te sluiten. Deze gebruiken we voor een toetsenbord, muis en WiFi dongle.

Verder zijn er nog 2 devices waarvan de zeppelin gebruik maakt:

- De camera laat toe foto's te nemen met een maximum resolutie van 5 MP. Hiermee kunnen we onder andere beelden maken van QR-codes. Daarnaast kan de camera video's maken met resoluties tot 1080p.
- De afstandssensor kan worden gebruikt om de afstand te meten tussen de zeppelin en de grond of muur. Het maximale bereik is 4m en het minimale bereik is 2cm.



Figuur 1: Raspberry Pi

Software

Zoals reeds vermeld, zal de software volledig in Java geschreven zijn. Deze keuze hebben we gemaakt omdat we al veel ervaring hebben met het schrijven van Java-programma's. Python was een andere mogelijke keuze. Er is veel voorbeeldcode in Python te vinden voor de Raspberry Pi, maar we hebben voor de functies die wij nodig hebben voldoende kennis en referenties in Java gevonden. Het leren van een extra taal zou erg tijdrovend in verhouding met de voordelen die het brengt.

Om de software te schrijven, gaan we op de laptop gebruik maken van de Eclipse IDE. We maken ook gebruik van de Netbeans IDE. Deze is door de visuele interface veel gebruiksvriendelijker om de GUI te ontwerpen. Hierdoor moeten we ons niet bezig houden met alle code voor de lay-out handmatig te schrijven. Voor het aansturen van de GPIO-pinnen gebruiken we Pi4J¹.

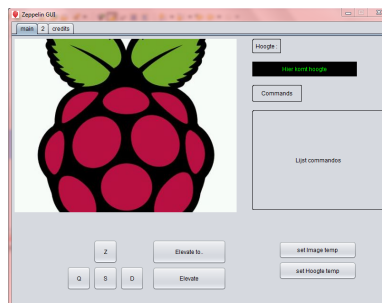
Op een client-pc kan de GUI(zie figuur 2 worden gestart. Hiermee kan de gebruiker de zeppelin aansturen via de pijltjestoetsen. Deze commando's worden doorgestuurd aan de zeppelin om te verwerken. Op de GUI kan de gebruiker informatie aflezen zoals de hoogte en afbeeldingen van de camera.

De communicatie tussen de GUI (client) en de Raspberry Pi (server) gebeurt via sockets. Dit is zoals een deur waarlangs objecten van klassen uitgewisseld worden. Deze stellen bijvoorbeeld commando's van de gebruiker voor of de status van de zeppelin die aan de GUI wordt doorgegeven. Concreet opent de server een socket op poort 6789 (de eerste 1024 poorten zijn gereserveerd voor services zoals http en ftp, we hebben een willekeurige poort hierachter gekozen). Hierop kan 1 client verbinden. Het zou mogelijk zijn meerdere clients toegang te geven, maar om conflicten (met pijltjestoetsen) te vermijden en om bandbreedte te besparen (bij het doorsturen van afbeeldingen), ondersteunt de server maar 1 client.

Om meerdere taken tegelijk te kunnen uitvoeren, maken we gebruik van threads. Zo kan de zeppelin tegelijk foto's nemen en de afstand tot de grond meten. Later zal worden beslist hoeveel gelijklopende threads er gebruikt worden.

Alle software behalve de GUI draait op de Raspberry Pi. Omdat de Raspberry Pi autonoom moet kunnen werken, laten we het grootste gedeelte van de software hierop draaien. Hier worden beslissingen genomen op basis van QR-codes, die uit afbeeldingen komen die de camera op regelde tijdstippen neemt. Elke opname van de camera wordt ook naar de GUI doorgestuurd. Het verwerken van de beelden zal gebeuren op de Raspberry Pi. Logisch gezien is het weergeven van de status van de zeppelin echter geen taak van de Pi zelf. Daarom draait de GUI op de client. Daarnaast wordt door deze keuze de cpu van de Pi minder zwaar belast.

De hoogte zal ook op bepaalde momenten opgemeten worden. Om een zo accuraat mogelijke waarde te krijgen, dient er een gemiddelde/mediaan genomen te worden van een aantal opeenvolgende meetresultaten. De hoogte wordt, net zoals de foto's, doorgestuurd naar de GUI.



Figuur 2: GUI

¹www.pi4j.com

Opmerking

Omdat we nog niet beschikken over alle nodige hardware, is het op dit moment onmogelijk om dieper in te gaan op de softwarematige aansturing van de motoren. In een latere fase van het project zal dit verder uitgewerkt worden.