Technisch document en test rapportage

Opdracht: een getest regelsysteem met decorators, en een python c++ binding

Naam: Martijn van de Koot

Student nummer: 1971895

Inleiding:

Dit document is ter ondersteuning van de opdracht. De opdracht komt vanuit het vak ATP, onderdeel van de cursus Innovatie. Als invulling op de opdracht is het volgende programma gemaakt:

Een digitale zonnevolger, met een gesimuleerde zonnesensor en een gesimuleerde magnetosensor. De zonnevolger beweegt zich binnen het programma door een gesimuleerde motor en geeft aan dat het verplaatsen richting de zon is voltooid door een gesimuleerde led. In het eerder geschreven “opzet en testplan” staan een aantal test beschreven die in het systeem aanwezig zijn. Niet alle beschreven tests zijn geïmplementeerd en de wel geïmplementeerde tests kunnen afwijken van het in eerste instantie opgestelde testplan. Ook zijn er meer tests toegevoegd dan er in het testplan beschreven waren. De rede waarom het testplan en de daadwerkelijk geschreven tests niet overeen komen is een verkeerde inschatting / een verkeerd beeld van de opdracht.

In dit document staan een aantal belangrijke keuzen en natuurlijk de tests beschreven.

Inhoud

[Toelichtingen op gemaakte keuzes 2](#_Toc133359402)

[getMagnetoSensorValues (readSensorValuesAndConvertToShort in c++) 2](#_Toc133359403)

[Test framework 2](#_Toc133359404)

[Test rapportage 3](#_Toc133359405)

[Unit tests: 3](#_Toc133359406)

[Integratie tests: 8](#_Toc133359407)

[Systeemtest: 11](#_Toc133359408)

# Toelichtingen op gemaakte keuzes

## getMagnetoSensorValues (readSensorValuesAndConvertToShort in c++)

De magnetosensor meet magnetische velden, en geeft deze terug in x, y en z richting. De sensor module heeft een 10 bit ADC, maar per richting twee 8 bit registers. Deze registers worden apart van elkaar uitgelezen en deze 2 waarden moeten dus nog omgezet worden tot één getal. De sensor kan positieve en negatieve waarden terug geven. Dit betekent dat er gewerkt wordt met signed en unsigned intergers. Dit heb ik geprobeerd in python werkend te krijgen maar dat lukte niet. Daarom heb ik ervoor gekozen deze functionele functie in c++ te schrijven en via Pybind11 direct de al tot integer omgezette waarden uit de c++ code te halen. Pybind11 zet een uint\_8t vanuit c++ om in een 32 bit signed integer.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Fig. 1, Mislukte poging signed en unsigned waarden in python te krijgen.

## Test framework

Als test framework heb ik de standaard unittest module uit python gebruikt. Tijdens de zelfstudie over testen kwam ik een aantal goede voorbeelden tegen die gebruik maken van deze module. Daarbij is de module makkelijk in gebruik door de ingebouwde assert functies en de mogelijkheid om automatisch alle tests uit te voeren. Voor deze test library zijn nog enorm veel alternatieven maar het enige alternatief wat ik overwogen heb is zonder framework te werken. Dit leek mij niet een goed plan omdat ik dan makkelijker fouten maak in het schrijven van de tests zelf.

# Test rapportage

## Unit tests:

De code bevat een aantal unit tests. Deze unit tests testen de werking van de zonnesensor simulator. De zonnesensor simulator is het belangrijkste onderdeel van het systeem want alle andere onderdelen werken aan de hand van de informatie die verkregen word uit de zonnesensor.

Afbeelding met tekst, envelop, visitekaartje

Automatisch gegenereerde beschrijvingDe test is opgedeeld in een aantal subtests, alle subtests testen een eigen deel van de functie.

Fig. 2,3D ontwerp zonnesensor

**Subtest1:**

Naam:

test\_creating\_cpp\_object

Beschrijving:

het testen van het importeren van de voor de zonnesensor simulatie benodigde c++ modules die in pybind11 omgezet zijn.

Deze test faalt als:

Pybind11 modules niet (goed) gecompileerd zijn, Pybind11 modules niet op het goede path staan, module namen of functie namen niet kloppen, modules niet op de juiste manier geïmporteerd zijn.

Deze test slaagt als:

Er in python, binnen de testcase, van alle toegevoegde Pybind11 modules een object gemaakt kan worden.

Verwacht resultaat: Verkregen resultaat:

Geslaagd. Geslaagd.

**Subtest2:**

Naam:

test\_zonne\_sensor\_input\_to\_value\_5degrees

Beschrijving:

Deze test voert roept de simulatie functie aan(c++ functie) met als hoek 5 graden azimut en 0 graden elevatie. De test bepaald of de gesimuleerde hoek (in grote schaal) klopt aan de hand van de terug gekregen zonnesensor waarden.

Extra informatie:

De simulatie code bevat om verschillende sensorwaarden terug te geven en eventuele afwijking na te bootsen een random generator. Dit kan er voor zorgen dat de return waarden tot 5 graden afwijken. Dit is de reden dat de test werkt met zijde +5 graden.

Deze test zakt als:

Blijkt dat de terug gekregen sensor waarden aangeven dat de zon op de verkeerde kant van de sensor schijnt en dus een verkeerde hoek simuleert. De zonnesensor simulator faalt waarden terug te geven aan de hand van de meegegeven hoeken.

Deze test slaag als:

Uit de terug gekregen waarden blijkt dat de zon op verwachtte zijde, de 0 graden zijde, van de sensor schijnt.

Verwacht resultaat: Verkregen resultaat:

Geslaagd. Geslaagd.

**Subtest3:**

Naam:

test\_zonne\_sensor\_input\_to\_value\_95degrees

Beschrijving:

Deze test voert roept de simulatie functie aan(c++ functie) met als hoek 95 graden azimut en 0 graden elevatie. De test bepaald of de gesimuleerde hoek (in grote schaal) klopt aan de hand van de terug gekregen zonnesensor waarden.

Extra informatie:

De simulatie code bevat om verschillende sensorwaarden terug te geven en eventuele afwijking na te bootsen een random generator. Dit kan er voor zorgen dat de return waarden tot 5 graden afwijken. Dit is de reden dat de test werkt met zijde +5 graden.

Deze test zakt als:

Blijkt dat de terug gekregen sensor waarden aangeven dat de zon op de verkeerde kant van de sensor schijnt en dus een verkeerde hoek simuleert. De zonnesensor simulator faalt waarden terug te geven aan de hand van de meegegeven hoeken.

Deze test slaag als:

Uit de terug gekregen waarden blijkt dat de zon op verwachtte zijde, de 90 graden zijde, van de sensor schijnt.

Verwacht resultaat: Verkregen resultaat:

Geslaagd. Geslaagd.

**Subtest4:**

Naam:

test\_zonne\_sensor\_input\_to\_value\_185degrees

Beschrijving:

Deze test voert roept de simulatie functie aan(c++ functie) met als hoek 185 graden azimut en 0 graden elevatie. De test bepaald of de gesimuleerde hoek (in grote schaal) klopt aan de hand van de terug gekregen zonnesensor waarden.

Extra informatie:

De simulatie code bevat om verschillende sensorwaarden terug te geven en eventuele afwijking na te bootsen een random generator. Dit kan er voor zorgen dat de return waarden tot 5 graden afwijken. Dit is de reden dat de test werkt met zijde +5 graden.

Deze test zakt als:

Blijkt dat de terug gekregen sensor waarden aangeven dat de zon op de verkeerde kant van de sensor schijnt en dus een verkeerde hoek simuleert. De zonnesensor simulator faalt waarden terug te geven aan de hand van de meegegeven hoeken.

Deze test slaag als:

Uit de terug gekregen waarden blijkt dat de zon op verwachtte zijde, de 180 graden zijde, van de sensor schijnt.

Verwacht resultaat: Verkregen resultaat:

Geslaagd. Geslaagd.

**Subtest5:**

Naam:

test\_zonne\_sensor\_input\_to\_value\_275degrees

Beschrijving:

Deze test voert roept de simulatie functie aan(c++ functie) met als hoek 275 graden azimut en 0 graden elevatie. De test bepaald of de gesimuleerde hoek (in grote schaal) klopt aan de hand van de terug gekregen zonnesensor waarden.

Extra informatie:

De simulatie code bevat om verschillende sensorwaarden terug te geven en eventuele afwijking na te bootsen een random generator. Dit kan er voor zorgen dat de return waarden tot 5 graden afwijken. Dit is de reden dat de test werkt met zijde +5 graden.

Deze test zakt als:

Blijkt dat de terug gekregen sensor waarden aangeven dat de zon op de verkeerde kant van de sensor schijnt en dus een verkeerde hoek simuleert. De zonnesensor simulator faalt waarden terug te geven aan de hand van de meegegeven hoeken.

Deze test slaag als:

Uit de terug gekregen waarden blijkt dat de zon op verwachtte zijde, de 270 graden zijde, van de sensor schijnt.

Verwacht resultaat: Verkregen resultaat:

Geslaagd. Geslaagd.

**Subtest6:**

Naam:

test\_zonne\_sensor\_input\_to\_value\_90degrees\_elevation

Beschrijving:

Deze test voert roept de simulatie functie aan(c++ functie) met als hoek 5 graden azimut en 90 graden elevatie. De test bepaald of de gesimuleerde hoek (in grote schaal) klopt aan de hand van de terug gekregen zonnesensor waarden.

Extra informatie:

De simulatie code bevat om verschillende sensorwaarden terug te geven en eventuele afwijking na te bootsen een random generator. Dit kan er voor zorgen dat de return waarden tot 5 graden afwijken. Dit is de reden dat de test werkt met zijde +5 graden.

Deze test zakt als:

Blijkt dat de terug gekregen sensor waarden aangeven dat de zon op de verkeerde kant van de sensor schijnt en dus een verkeerde hoek simuleert. De zonnesensor simulator faalt waarden terug te geven aan de hand van de meegegeven hoeken.

Deze test slaag als:

Uit de terug gekregen waarden blijkt dat de zon op verwachtte zijde, de bovenste zijde, van de sensor schijnt.

Verwacht resultaat: Verkregen resultaat:

Geslaagd. Geslaagd.

**Subtest7:**

Naam:

test\_zonne\_sensor\_input\_to\_value\_handle\_negative\_values

Beschrijving:

Deze test voert roept de simulatie functie aan(c++ functie) met als hoek -50 graden azimut en -90 graden elevatie. De test bepaald of de simulator om kan gaan met negatieve waarden door te checken of de functie een error terug geeft.

Extra informatie:

Wanneer deze test slaagt zal de functie wel sensorwaarden terug geven. Deze sensorwaarden zullen niet kloppen maar dit betekent wel dat het programma in ieder geval verder kan en niet het systeem plat zal gooien.

Deze test zakt als:

De functie crasht of zich onveilig gedraagt.

Deze test slaag als:

Als de functie compleet uitgevoerd is en waarden terug geeft.

Verwacht resultaat: Verkregen resultaat:

Geslaagd. Geslaagd.

**Subtest8:**

Naam:

test\_zonne\_sensor\_input\_to\_value\_handle\_extreme\_positive\_values

Beschrijving:

Deze test voert roept de simulatie functie aan(c++ functie) met als hoek 10000 graden azimut en 10000 graden elevatie. De test bepaald of de simulator om kan gaan met extreem hoge waarden door te checken of de functie een error terug geeft.

Extra informatie:

Wanneer deze test slaagt zal de functie wel sensorwaarden terug geven. Deze sensorwaarden zullen niet kloppen maar dit betekent wel dat het programma in ieder geval verder kan en niet het systeem plat zal gooien.

Deze test zakt als:

De functie crasht of zich onveilig gedraagt.

Deze test slaag als:

Als de functie compleet uitgevoerd is en waarden terug geeft.

Verwacht resultaat:

Geslaagd.

Verkregen resultaat:

Geslaagd.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Fig. 3, Testresultaat

## Integratie tests:

Om zeker te zijn van de werking van beide sensoren en de functies waarmee sensor data verwerkt wordt is de code voor het uitlezen, simuleren en verwerken van uitgelezen data verwerkt in twee integratie tests. Voor beide sensoren één test. Het testen begint met één korte test van de implementatie van Pybind11

**Test1:**

Naam:

test\_cpp\_includes

Beschrijving:

Aan de hand van deze test is te bepalen of de Pybind11 modules correct werken. Dit is van belang aangezien alle code binnen de rest van de tests gebruik maakt van deze modules.

Deze test faalt als:

Pybind11 modules niet (goed) gecompileerd zijn, Pybind11 modules niet op het goede path staan, module namen of functie namen niet kloppen, modules niet op de juiste manier geïmporteerd zijn.

Deze test slaagt als:

Er in python, binnen de testcase, van alle toegevoegde Pybind11 modules een object gemaakt kan worden.

Verwacht resultaat:

Geslaagd

Verkregen resultaat:

Geslaagd

**Test2**

Naam:

test\_zonne\_sensor\_normal\_values

Beschrijving:

Aan de hand van het resultaat van deze test is af te lezen of de simulatie van de zonnesensor geslaagd is. Of de uit de simulatie verkregen sensorwaarden binnen functies verwerkt zijn tot een kloppende uitkomende hoek.

Extra informatie:

Er wordt een azimut en elevatie hoek meegegeven aan de simulatie. Deze hoek wordt verwerkt tot sensorwaarden. Deze sensorwaarden worden door functies weer terug omgerekend tot een hoek. De simulator maakt gebruik van random om een afwijking te creëren in de uitkomst. Deze afwijking is acceptabel tussen -12 graden en +12 graden van de meegegeven hoeken(-6 en +6 voor de azimut hoek).

De test bestaat uit twee functies. Functie 1 simuleert de hoeken en berekent de hoek aan de hand van de gesimuleerde waarden. Functie 2 is de test. De test roept de eerste functie aan met alle waarden tussen 0 en 360. De resultaten worden in de tweede functie vergeleken met de invoerwaarden(hierbij wordt rekening gehouden met de afwijking). Het resultaat wordt binnen de functie opgeslagen in een lijst.

De test zakt als:

Blijkt dat er in de lijst met test uitkomsten een gefaalde testwaarde aanwezig is. (Er een niet kloppende uitkomst is in vergelijking met de invoer)

De test slaagt als:

Alle geteste waarden na simulatie en verwerking binnen de marge van 12 graden blijken te liggen in vergelijking met de invoerhoek.

Verwacht resultaat:

Geslaagd

Verkregen resultaat:

Gezakt

Evaluatie:

De test is gezakt omdat er gesimuleerde, berekende waarden bij zitten die niet binnen de marge van 12 graden afwijking zitten vergeleken met de invoerwaarde. Hiervoor zijn verschillende mogelijkheden, een mogelijkheid kan zijn dat er een berekeningsfout zit in de verwerking van de sensordata of sensorsimulatie. Een andere mogelijkheid is dat de random factor zo afgesteld staat dat er een afwijking gecreëerd wordt van meer als 12 graden.

Mogelijke oplossingen:

* De random factor bijstellen
* De simulator (verder) testen en debuggen
* De functies voor sensorverwerking testen en debuggen

Afbeelding met tekst

Automatisch gegenereerde beschrijving Fig. 4, Testresultaat

**Test3**

Naam:

test\_magneto\_sensor

Beschrijving:

Aan de hand van het resultaat van deze test is af te lezen of de simulatie van de magnetosensor geslaagd is. Of de uit de simulatie verkregen sensorwaarden binnen functies verwerkt zijn tot een kloppende hoek.

Extra informatie:

Er wordt een azimut hoek meegegeven aan de simulatie. Deze hoek wordt verwerkt tot sensorwaarden. Deze sensorwaarden worden door functies weer terug omgerekend tot een hoek. De simulator maakt gebruik van random om een afwijking te creëren in de uitkomst. Deze afwijking is acceptabel tussen -5 graden en +5 graden van de meegegeven hoek.

De test bestaat uit twee functies. Functie 1 simuleert de hoek en berekent de hoek aan de hand van de gesimuleerde waarden. Functie 2 is de test. De test roept de eerste functie aan met alle waarden tussen 0 en 360 (een complete cirkel). De resultaten worden in de tweede functie vergeleken met de invoerwaarden (hierbij wordt rekening gehouden met de afwijking). Het resultaat wordt binnen de functie opgeslagen in een lijst.

De test zakt als:

Blijkt dat er in de lijst met test uitkomsten een gefaalde testwaarde aanwezig is. (Er een niet kloppende uitkomst waarde is in vergelijking met de invoerwaarde)

De test slaagt als:

Alle geteste waarden na simulatie en verwerking binnen de marge van 5 graden blijken te liggen in vergelijking met de invoerhoek.

Verwacht resultaat:

Geslaagd

Verkregen resultaat:

Gezakt

Evaluatie:

De test is gezakt omdat er gesimuleerde / berekende waarden bij zitten die niet binnen de marge van 5 graden afwijking zitten vergeleken met de invoerwaarde. Hiervoor zijn verschillende mogelijke redenen, een mogelijke reden kan zijn dat er een berekeningsfout zit in de verwerking van de sensordata of in de sensorsimulatie. Een andere mogelijkheid is dat de random factor zo afgesteld staat dat er een afwijking gecreëerd wordt van meer als 5 graden.

Mogelijke oplossingen:

* De random factor bijstellen
* De simulator (verder) testen en debuggen
* De functies voor sensorverwerking testen en debuggen
* Een grotere afwijking tolereren

Afbeelding met tekst

Automatisch gegenereerde beschrijving



Fig. 5, testresultaat

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Fig. 6, testresultaat

## Systeemtest:

Naam:

test\_sun\_pointer

Beschrijving:

Deze test test het complete systeem door de simulatie te runnen. Aan de hand van deze test is te zien of het systeem op minimaal 80% van de tijd de tijd de zon met een maximale afwijking van 3 graden gevolgd heeft.

Extra informatie:

Deze test is belangrijk omdat het bewijst dat het systeem nauwkeurig is (80% van de tijd de juiste positie aanwijst)

De test zakt als:

Het systeem crasht, de zonnewijzer meer als 20% van de tijd niet de juiste richting aangeeft.

De test slaagt als:

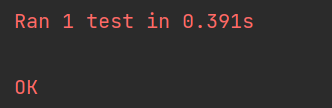
Het systeem de complete mainloop heeft afemaakt en daarbij een minimale nauwkeurigheid heeft van 80% van de tijd(cycles) de goede richting aangeeft (azimut en elevatie) met een maximale afweiking van 3 graden.

Verwacht resultaat:

Geslaagd

Verkregen resultaat:

Geslaagd

Fig. 7, testresultaat

Afbeelding met grafiek

Automatisch gegenereerde beschrijving

Fig. 8, Output grafiek systeemtest(simulatie)

Toelichting:

Groen = azimut hoek zon

Rood = elevatiehoek zon

Blauw = positie zonnewijzer azimut

Oranje = positie zonnewijzer elevatie