

IPASS - PvA

Stijn van Wijk, 1744230

Stijn.vanwijk@student.hu.nl

13-05-2019

Beschrijving van project

Hardware

Voor mijn IPASS-project, ga ik de volgende hardware gebruiken:

Hardware	Arduino Due
Beschrijving	Dit is de hart van het project hierop ga ik mijn programma runnen, die gebruik maakt van de library die ik geschreven heb.
URL	http://technische-informatica.nl/ti-lab-shop/Arduino%20Due.html
Datasheet	http://ww1.microchip.com/downloads/en/devicedoc/atmel-11057-32-bit-cortex-m3-microcontroller-sam3x-sam3a_datasheet.pdf

Hardware	MAX7219 met 8x8 LED matrix
Beschrijving	<p>De MAX7219 is een leddriver, waaraan een 8x8 LED matrix is verbonden. Ik ben van plan om voor deze leddriver een library te schrijven.</p> <p>De leddriver kan met een 16 bit input, 64 individuele leds aansturen. Hij maakt gebruik van persistence of vision, om onze hersenen wijs te maken dat een ledje de hele tijd aanstaat.</p> <p>De chip verwacht een 16-bit input, als je meer bitjes als input geeft dan shift je de eerste bitjes weg uit de chip via de data-out (shift register). Op deze manier kan je de chip daisy chainen.</p>
URL	http://technische-informatica.nl/ti-lab-shop/mod-max7219-8x8-b.html
Datasheet	https://datasheets.maximintegrated.com/en/ds/MAX7219-MAX7221.pdf

Hardware	MPU6050
Beschrijving	De MPU6050, is een acceleratie & gyro sensor. Ik wil met mijn project gebruik maken van de gyrosensor. Je kan met de MPU6050 communiceren met I2C. De MPU6050 heeft een adres op de I2C-bus, en door data uit te lezen uit registers kan je informatie vergaren.
URL	http://technische-informatica.nl/ti-lab-shop/sensor-acceleration-mpu6050.html
Datasheet	https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf

Er zijn in afgelopen jaren, meer IPASS-projecten geweest die gebruik maken van de MAX7219 of de MPU6050. Er zijn dus ook voorbeelden van libraries voor deze hardware. De volgende projecten hebben een library geschreven voor de MAX7219:

<http://technische-informatica.nl/ipass/Bart%20Commandeur//index.html>

<http://technische-informatica.nl/ipass/Mike%20Hilhorst//index.html>

<http://technische-informatica.nl/ipass/Nick%20Goris//index.html>

<http://technische-informatica.nl/ipass/Rik%20Ruttenberg//index.html>

De volgende projecten hebben een library geschreven voor de MPU6050:

<http://technische-informatica.nl/ipass/Brian%20Duncker//index.html>

<http://technische-informatica.nl/ipass/Mart%20van%20der%20Ham//index.html>

Een project dat gebruikt maakt van zowel de MPU6050 als de MAX7219, is het volgende project:

<https://www.instructables.com/id/LED-Sand-in-a-Box/>

Aan de hand van deze demoprojecten, kan ik een idee krijgen welke functionaliteiten mijn library kan bevatten.

Library

Eerste stap

Ik wil dus een library schrijven voor de MAX7219 LED driver. Na overleg van Wouter, ga ik als interface class `hwlib::window` gebruiken. De library die ik ga schrijven moet dus de functies die in `hwlib::window` staan implementeren voor de MAX7219.

Ik hoef dus niet na te denken aan een interface class. Maar wat het voor mij uitdagend maakt is hoe krijg ik `hwlib::window` werkend met de MAX7219. Op het moment van schrijven vind ik het lastig om te bepalen hoe en waar ik moet beginnen. Hoe vertaal ik de `hwlib::window::write/flush/clear` functies naar iets wat de MAX7219 begrijpt.

Tweede stap (optioneel)

Als ik een succesvolle library heb geschreven en mijn applicatie werkend heb, ben ik van plan nog een kleine library voor de MPU-6050 schrijven. Het enige waar ik op dit moment geïnteresseerd in ben is de gyroscope data. Ik verwacht dat hierdoor mijn library kleiner kan zijn, dan één die ik kan vinden op het internet. Dit is handig, want de ruimte is immers beperkt op een Arduino due.

Applicatie

Eerste stap

Als applicatie wil ik een soort physics simulator schrijven. Door naar de gyro data te kijken van de MPU6050, bepaal ik de richting van de gravitatie. Het doel van mijn applicatie is om op een 2D vlak water te simuleren. Als je de ledmatrix kantelt, zal het water ook bewegen in dezelfde richting als de kanteling.

Ik denk dat de volgende onderdelen mijn applicatie uitdagend maken:

- Hoe werkt de gravitatie op een object in mijn simulatie? Is de versnelling altijd gelijk, of vergroot je die?
- Hoe gedraagt een object zich als die collide met een ander object?
- Als een object niet verder naar beneden kan bewegen, richting het middenpunt/zwaartekracht, omdat er al andere objecten aanwezig zijn, hoe zorg ik ervoor dat ze stoppen met bewegen richting het zwaartepunt?

Ik denk dat deze applicatie een mooie uitdaging vormt.

Tweede stap (optioneel)

Als ik na het schrijven van twee libraries en een werkende applicatie heb. Zou ik mijn applicatie, de physics simulator nog kunnen vertalen naar een library. Ik verwacht dat ik niet ga halen, maar stel je voor de ik tijd over heb ben ik van plan hiermee aan de slag te gaan.

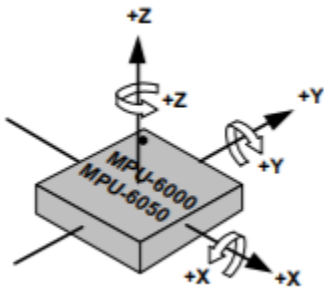
Risicobeheersing

In mijn library voor de MAX7219, zie ik de volgende versies voor me:

1. Ik kan 1 MAX7219 (met een 8x8 LED matrix) aansturen, en die gedraagt zich als een `hwlib::window` van 8 bij 8.
2. Ik kan de MAX7219 daisy-chainen. De daisy-chain is dan alleen horizontaal. Ik kan nog geen matrix aanmaken van MAX7219 drivers. Door de horizontale daisy-chain kan ik een `hwlib::window` aanmaken van bijvoorbeeld 16x8.
3. Ik kan een matrix aanmaken van MAX7219, hierdoor is het mogelijk om een window te maken van bijv 16x16. Door een matrix te hebben van MAX7219 van 2x2.
4. Ik kan een window aanmaken die NIET een volledige MAX7219 inneemt. Ik kan dus zeggen dat het een window is van bijv. 4x4 of 16x9. Mijn library weet dan nog steeds hoe die zich moet gedragen.
Dit is mogelijk helemaal geen andere versie, maar op dit moment weet ik nog niet hoe de library er exact uit komt te zien.

In mijn applicatie zie ik ook meerdere versies:

1. De objecten in de simulatie volgen de zwaartekracht, ze reageren niet op collisions met andere objecten. De objecten vallen niet uit het scherm!
De draaing is in dit geval maar in 1-axis.



Het ligt eraan hoe ik de MPU6050 monteer, maar laten we in mijn voorbeeld onderstellen dat de Y-as naar boven is gericht (90° op de horizon). In de eerste versie zullen de objecten alleen reageren in bewegingen op de z-as.

2. Ik voeg collision detection toe.
3. De objects in mijn simulatie weten hoe ze reageren op een collision.
4. De objecten weten wanneer ze niet verder kunnen bewegen omdat er andere objecten in de weg zitten.

Ik denk dat ik mijn IPASS-project genoeg heb opgedeeld om te weten wat ik ieder geval wil opleveren. Ik hoop dat mijn PvA een duidelijk beeld geeft van wat ik als basis zie, en wat ik als uitbreiding op de basis zie.