

# Plan voor IPASS

## Gegevens

**Naam:** Youri Mulder  
**Studentnummer:** 1716390  
**E-mail adres:** [Youri.y.mulder@student.hu.nl](mailto:Youri.y.mulder@student.hu.nl)  
**Datum van document:** 8/5/2018

## Beschrijving project

### Mijn project

Voor mijn IPASS wil ik graag een alarmsysteem maken. Het alarm systeem wordt opgebouwd door een combinatie van verschillende modules/chips. Het belangrijkste aspect van dit alarmsysteem moet worden dat het werkt doormiddel van RFID en een Real time clock. Daarom wil ik voor beide chips een library te schrijven. Ook wil ik niet iedereen dezelfde rechten geven. De eigenaar is alleen in staat om het alarm uit te zetten wanneer hij is af gegaan. De andere gebruikers kunnen alleen het alarm aan of uit zetten mits het alarm niet is afgegaan.

### Welke hardware ga je gebruiken?

Naast de arduino due wil ik gebruik gaan maken van hardware. De informatie over deze hardware staat hieronder

#### *RC-522 (RFID reader/writer)*

##### Wat is het voor een chip?

De RC522 is een contactloze lezer en schrijver. Die gemaakt is om te kunnen communiceren met ISO/IEC 14443A/MIFARE kaarten. Deze kaarten worden meegeleverd bij de verzending. Je kan op verschillende manieren deze chip benaderen namelijk: SPI, UART, I2C. Ik verwacht de I2C modus te gebruiken, omdat ik ook andere chips ga gebruiken die alleen I2C ondersteunen. Als ik geen I2C bus nodig had voor de andere chips, had ik gekozen voor SPI aangezien dit een stuk snellere gegevensuitwisseling heeft.

##### Waar ga ik de chip voor gebruiken?

Niet iedereen mag zomaar een alarm uitzetten. Daarom ga ik deze chip gebruiken. Door middel van deze chip kan ik goed bijhouden wie het alarm erop zet of eraf haalt. Het zou ook mogelijk zijn om een code hiervoor te gebruiken, maar persoonlijk vindt ik een kaartje voor een lezer houden toch wat gebruiksvriendelijker.

##### Datasheet

<https://www.elecrow.com/download/MFRC522%20Datasheet.pdf>  
<https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf>

##### Verkrijgbaar

<https://goo.gl/PRBCGx> (Aliexpress)

##### Voorbeeld

<https://www.brainy-bits.com/card-reader-with-an-arduino-rfid-using-the-rc522-module/>

### DS3231 AT24C32 (Real time clock/ RTC)

Wat is het voor een chips?

De DS3231 is een real time clock(RTC) die je kan aansturen met gebruik van het I2C protocol. Deze module bevat ook een serial EEPROM chip namelijk: AT24C32. Deze EEPROM is ook aan te sturen via I2C. De EEPROM heeft een grootte van 32768 bits. Ook heeft deze chip een thermometer ingebouwd. Hierdoor kan je meten hoe warm het is in de kamer.

Waar ga ik de chip voor gebruiken?

Deze chip ga ik gebruiken om de tijd weer te geven op een OLED scherm en een log bij te houden wanneer en door wie het alarm voor het laatst aangezet of is uitgezet is. Deze gegevens zou ik in de EEPROM van de module kunnen schrijven. Ook ga ik de temperatuur van de kamer op een OLED scherm afbeelden.

Datasheet

<https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>

Verkrijgbaar

<https://goo.gl/B6GRK1> (Aliexpress) reeds binnen

Voorbeeld project

<https://howtomechatronics.com/tutorials/arduino/arduino-ds3231-real-time-clock-tutorial/>

### HC-SR501 (Beweging sensor)

Wat is het voor een chip?

HC-SR501 is een beweging sensor. De module maakt gebruik van infrarood die je kan instellen op gevoeligheid en vertraging. Ook kan je kiezen of je een keer een output geeft als er iets is gedetecteerd of constant.

Waar ga ik de chip voor gebruiken?

Een beweging sensor is cruciaal bij een alarm. Je moet natuurlijk wel kunnen detecteren of er iemand binnen is als het alarm aan staat. Daarom heb ik gekozen voor deze module. Want het biedt de mogelijkheden die ik nodig heb zonder veel zelf eraan te hoeven knutselen en programmeren.

Datasheet

<https://www.mpja.com/download/31227sc.pdf>

Verkrijgbaar

<https://goo.gl/Kp8n2t> (Aliexpress) reeds binnen

Voorbeeld project

<http://henrysbench.capnfatz.com/henrys-bench/arduino-sensors-and-input/arduino-hc-sr501-motion-sensor-tutorial/>

### *OLED display I2C*

Wat is het voor een chip?

Het OLED display is een I2C chip waar je een pixel op een bepaalde kleur kan zetten. In mijn geval is dit wit. Je kan hier allerlei dingen op afbeelden. Als het maar binnen de resolutie van 128x64 past.

Waar ga ik de chip voor gebruiken?

Dit display wil ik gaan gebruiken om de tijd, status van het alarm, RFID meldingen, temperatuur en nog wat andere gegevens die mijn modules doorgeven op af te beelden.

Datasheet

[https://www.buydisplay.com/download/manual/ER-OLED0.96\\_Series\\_Datasheet.pdf](https://www.buydisplay.com/download/manual/ER-OLED0.96_Series_Datasheet.pdf)

Verkrijgbaar

<https://goo.gl/nw8CvY> (Aliexpress) reeds binnen

Voorbeeld project

<https://startingelectronics.org/tutorials/arduino/modules/OLED-128x64-I2C-display/>

### *Actieve Buzzer*

Wat is het voor een chip?

Deze buzzer kan verschillende geluiden produceren.

Waar ga ik de chip voor gebruiken?

Ik wil als het alarm af gaat een geluid produceren die aangeeft.

Datasheet

[https://www.buydisplay.com/download/manual/ER-OLED0.96\\_Series\\_Datasheet.pdf](https://www.buydisplay.com/download/manual/ER-OLED0.96_Series_Datasheet.pdf)

Verkrijgbaar

<https://goo.gl/JZz86L> (Aliexpress) reeds binnen

Voorbeeld project

<https://startingelectronics.org/tutorials/arduino/modules/active-buzzer/>

## Library

### *Welke library ga je maken?*

Ik ben van plan om twee libraries te maken. Ik wil de libraries gaan maken voor de DS3231(RTC) en RC-522(RFID). Ik verwacht de meeste uitdaging te hebben in het maken voor de library van de RFID. Beide libraries worden geplaatst op github. De source-code kun je vinden op:

[https://github.com/YouriMulder/IPASS\\_2018](https://github.com/YouriMulder/IPASS_2018)

### *Klassen en methode*

#### DS3231 (RTC)

Dit is een klasse voor het aansturen voor de chips DS3231 en AT24C32 die ingebouwd zit in deze module.

#### Methodes

Deze klasse heeft verschillende methodes nodig. Veel methodes verkrijgen data uit een register. Daarom wil ik een paar methodes maken die een register aanspreken. Als het blijkt dat het makkelijker is om deze methode op te nemen in een andere klasse, omdat ik ze dan kan hergebruiken ga ik dat doen. De methode zijn:

- `Void setRegister(uint8_t chipRegAddress);`
- `UInt8_t getByteFromRegister(uint8_t chipRegAddress);`
- `UInt8_t getDECFromBCDRegister(uint8_t chipRegAddress);`

Door deze methodes kan ik makkelijk de data uit de registers halen. Een aantal methode die ik makkelijk hierdoor kan maken zijn:

- `UInt8_t getCurrentSeconds();`
- `UInt8_t getCurrentMinutes();`
- `UInt8_t getCurrentHours();`
- Etc.

Naast de methodes om de tijd te verkrijgen moet ik ze ook kunnen zetten.

- `setByteInRegister(uint8_t chipRegAddress, uint8_t byte);`
- `setCurrentSeconds(uint8_t newSeconds);`
- `setCurrentMinutes(uint8_t newMinutes);`

Omdat deze module ook nog een termostaat bevat zijn de volgende methodes ook handig.

- `GetCurrentTemperatureCelsius();`
- `GetCurrentTemperatureFahrenheit();`

De AT24C32 is een deel van de module. Hiervoor heb ik geen andere methodes nodig, omdat ik de `setByteInRegister` en `getByteFromRegister` al heb. Het is alleen van belang dat ik het I2C adres verander.

## Timestamp

Omdat de DS3231 de tijd bijhoudt is het makkelijk als je een bepaalde timestamp kan opslaan en bewerken. Hiervoor ga ik een klasse gebruiken die Timestamp heet. In de klasse kan je de seconden, minuten, etc. opslaan en eventueel naar een andere tijdzone omzetten. De klasse DS3231 krijgt bijhorende methodes om direct in een timestamp op te slaan.

## RC-522 (RFID)

Voor deze chip heb ik een minder goed idee hoe ik dat wil gaan verwezenlijken. Ik heb in iedergeval methodes nodig die registers uit kunnen lezen op de goede manier. Ik verwacht dat di er ongeveer hetzelfde uit gaan zien als bij de DS3231 (RTC).

- `Void setRegister(uint8_t chipRegAddress);`
- `Uint8_t getByteFromRegister(uint8_t chipRegAddress);`
- `Uint8_t getDECFromBCDRegister(uint8_t chipRegAddress);`

Daarnaast heb ik methodes nodig die kijken of er een kaart gepresenteerd wordt en of deze kaart bekend is, omdat er niet zomaar een willekeurige kaart het alarm mag uitzetten.

- `bool/uint8_t cardPresented();`
- `uint8_t/uint16_t readCard();`
- `bool validCard();`
- `bool unknownCard();`

Naast het lezen van kaarten is het belangrijk nieuwe kaarten toe te kunnen voegen. Deze methodes verwacht ik zo te gaan maken:

- `void addCard(uint8_t/uint16_t newCard);`

Dit waren een aantal voorbeelden van mijn verwachting hoe deze library eruit gaat zien. Naast deze methodes moeten er ook een aantal private methodes komen die het bit shiften en bit checken regelen van de kaarten, maar ik heb nog geen idee hoe dit eruit gaat zien.

## Welke applicatie ga je maken?

### *Hoe heet de applicatie?*

De applicatie heet house shield, omdat het je huis of bedrijf beveilgd tegen ongewenste bezoekers.

### *Wat doet de applicatie?*

De applicatie beschermd je huis. Het zorgt ervoor dat ongewenste bezoekers worden afgeschrikt als het alarm af gaat.

### *Wat maakt de applicatie moeilijk/uitdagend?*

Ik verwacht dat het maken van de libraries een uitdaging wordt, omdat het gebruik maakt van I2C en veel bit shiften. Naast de libraries wordt het ook een klus om de verschillende modules en chips aan elkaar te koppelen tot een kloppend geheel, omdat een arduino due niet alle resources bevat die een moderne PC heeft.

## Risicobeheersing project

### *Welke versies zie je in de library en applicatie?*

In de libraries zie ik veel verschillende versies. Je kan elke aanpassing die je doet opslaan en behouden doormiddel van versie beheer(git en GitHub). Door het gebruik van git en GitHub is de kans nihil om je source-code kwijt te raken. Elke aanpassing die ik doe aan de library ga ik opslaan, omdat elk beetje werkende code meewerkt aan het eindresultaat. Als het voorkomt dat er een fout wordt gemaakt in nieuw geschreven code is het makkelijk terug te halen door git en GitHub.

Nadat ik de libraries heb gemaakt is het tijd voor de applicatie. De verschillende versies in de applicatie worden ook naar een repository geschreven. Als er iets gelukt is gaat er direct een commit en push naar deze repo.

### *Hoe ga je toch iets opleveren als het tegenzit?*

Als het niet lukt om de libraries als een geheel tot een applicatie te maken zorg ik ervoor dat de libraries individueel goed werken met een kleine demo. Als er een stuk van een library wegvalt is het mogelijk om een kijkje te nemen in een al bestaande library en dat over te nemen mits ik dat vermeld in de code. De losse delen zijn allemaal goed te vervangen doormiddel van een kijkje in een voorbeeld project, ook al verwacht/wil ik daar geen gebruik van gaan maken.

De belangrijkste beslissingen die ik ga maken wil ik opschrijven in een bestandje met motivatie erbij. Als ik inzie dat ik verkeerde keuzes heb gemaakt kan ik als de code niet goed werkt een document inleveren met de fouten die ik heb gemaakt en eventuele keuzes die ik zou veranderen in een volgend project, want in het bedrijfsleven worden er ook wel eens fouten gemaakt. Deze fouten willen bedrijven natuurlijk zo liefst mogelijk onder de tafel schuiven, maar persoonlijk vind ik het veel belangrijker te leren van de fouten die je hebt gemaakt en met deze kennis een volgend project naar succes helpen.