

# RFID-Eindproject

## Inleiding

Het RFID-eindproject gaat over het implementeren van een administratiesysteem voor het autoriseren van RFID-tags met behulp van de Arduino. Het systeem dat voor deze opdracht geïmplementeerd moet worden moet tenminste:

- RFID-tags kunnen uitlezen met behulp van een Arduino en een RFID-lezer;
- de UID van een RFID-tag beveiligd naar een computer kunnen versturen;
- de UID van een RFID-tag kunnen vergelijken met opgeslagen UIDs in een database om te bepalen of deze geautoriseerd is;
- de UID van een RFID-tag opslaan in of verwijderen uit de database.

## Arduino

Allereerst zal er een programma geïmplementeerd moeten worden voor de Arduino, waarmee het identificatienummer (UID) van een RFID-tag uitgelezen kan worden. Voor de implementatie van dit programma kan gebruik worden gemaakt van de Arduino programmeeromgeving.

Om met behulp van een Arduino RFID-tags te kunnen lezen, is een uitbreidingsmodule nodig. Hiervoor is de Mifare MFRC522 RFID Reader/Writer aangeleverd.

## Installatie software

Om het programma voor de Arduino te implementeren, kan gebruik gemaakt worden van de programmeeromgeving die beschikbaar is op [1]. Voor een tutorial over hoe deze omgeving te installeren is, zie [2].

## Mifare MFRC522 RFID Reader/Writer

De MFRC522 is een module die Mifare tags kan lezen en schrijven. De module kan op de Arduino aangesloten worden met behulp van een breadboard. De pin-layout is als volgt:

MFRC522 pin	Arduino pin
RST	9
SDA	10
MOSI	11
MISO	12
SCK	13
GND	GND
3.3V	3.3V

## Libraries

Voor het lezen van de UID van een tag kunnen de libraries worden gebruikt die via BlackBoard zijn aangeleverd (de mappen RFID en MFRC522). Hoe deze libraries gebruikt kunnen worden is te vinden op [3]. Voor de implementatie van de Arduino kan gebruik worden gemaakt van de functies van de RFID-library. Deze maakt op zijn beurt gebruik van de functies van de MFRC522-library, die afkomstig is van [4], en verschaft een rechtstreekse interface tussen de Arduino en de RFID-lezer. Het is dus nuttig om de functies van RFID te begrijpen, maar de functies van MFRC522 hoeven niet in detail bestudeerd te worden.

Om de libraries te gebruiken moet in een Arduino-sketch niet alleen "RFID.h" geïmporteerd worden, maar ook de libraries waar deze zelf gebruik van maakt.

**NB:** Deze libraries zijn bedoeld als hulpmiddel. Het is toegestaan om hier wijzigingen in aan te brengen als je hebt besloten om het systeem op een andere manier te implementeren.

## Administratiesysteem

Daarnaast moet het administratiesysteem geïmplementeerd worden, waarvoor bijvoorbeeld Python gebruikt kan worden. Het administratieprogramma moet voor een zekere UID kunnen bepalen of deze geautoriseerd is. Dit kan met behulp van een database die een tabel bevat met daarin alle geautoriseerde UIDs. Daarnaast moet het administratiesysteem ook de mogelijkheid bieden om tags toe te voegen aan de database of daaruit te verwijderen.

### Database

Met behulp van SQLite kan vanuit Python een database gecreëerd en gebruikt worden, die de UIDs bevat van alle geautoriseerde tags. Het administratiesysteem dat geïmplementeerd zal worden moet dus de mogelijkheid bieden om hier tags aan toe te voegen (en te verwijderen). Wanneer vervolgens gecontroleerd moet worden of een tag geautoriseerd is, kan de UID van die tag vergeleken worden met de entries van de database.

Om SQL te gebruiken in Python moet de SQLite3 command line tool geïnstalleerd worden vanaf [5]. Raadpleeg een tutorial voor meer informatie over het gebruik van SQLite in Python, bijvoorbeeld [6] of [7].

**NB:** In paragraaf 111 zal meer informatie over SQL gegeven worden.

## Communicatie

Om dit geheel te kunnen implementeren, is het natuurlijk nodig dat de Arduino en het administratiesysteem met elkaar kunnen communiceren. Hiervoor zal een communicatieprotocol geïmplementeerd moeten worden. De communicatie tussen de Arduino en Python is mogelijk door gebruik van PySerial, wat gedownload kan worden via [8]. Voor meer informatie over hoe communicatie opgezet kan worden tussen Arduino en Python, zie bijvoorbeeld [9] of [10] en de PySerial API die te vinden is via [11].

## Autorisatie

Vervolgens is het ook noodzakelijk dat een tag daadwerkelijk geautoriseerd kan worden. Als een tag zijn UID ongecodeerd verstuurd, kan een derde partij deze natuurlijk af luisteren en vervolgens zelf gaan gebruiken. Om dit te voorkomen moet een authenticatie-algoritme geïmplementeerd worden, bijvoorbeeld een vorm van challenge-response authenticatie [12]. Omdat de RFID-library de UID van een pasje oplevert als het datatype `uint32_t`, kan hier makkelijk mee gerekend worden.

**NB:** Het datatype `uint32_t` is precies groot genoeg om de UID in te bevatten van de pasjes die bij de opdracht geleverd worden. Mogelijk moet je hier rekening mee houden in je programma, als je berekeningen met de UID uitvoert.

## Uitbreidingen

Wanneer de basisfunctionaliteit van het programma geïmplementeerd is, is het natuurlijk mogelijk om hier nog enkele uitbreidingen aan toe te voegen. Voorbeelden daarvan zijn:

- een GUI implementeren;
- de opgeslagen UIDs in de database beveiligen;
- informatie naar een tag schrijven en van een tag lezen;
- etcetera.

## Benodigdheden

### **Aangeleverde bestanden**

- RFID-library met rfid.cpp en rfid.h
- MFRC522-library met MFRC522.cpp en MFRC522.h

### **Hardware**

- Laptop
- Arduino, breadboard en kabels
- Mifare MFRC522 RFID Reader/Writer inclusief tags

### **Software**

- Arduino programmeeromgeving
- Python
- PySerial
- SQLite3 command line tool

### **Referenties**

- [1] <http://arduino.cc/en/Main/Software>
- [2] <http://arduino.cc/en/Guide/HomePage>
- [3] <http://arduino.cc/en/Guide/Libraries>
- [4] <https://github.com/miguelbalboa/rfid>
- [5] <http://www.sqlite.org/>
- [6] <https://docs.python.org/3/library/sqlite3.html>
- [7] <http://zetcode.com/db/sqlitepythontutorial/>
- [8] <http://pyserial.sourceforge.net/>
- [9] <http://playground.arduino.cc/interfacing/python>
- [10] <http://www.instructables.com/id/Arduino-and-Python/>.
- [11] [http://pyserial.sourceforge.net/pyserial\\_api.html](http://pyserial.sourceforge.net/pyserial_api.html)
- [12] [http://en.wikipedia.org/wiki/Challenge%E2%80%93response\\_authentication](http://en.wikipedia.org/wiki/Challenge%E2%80%93response_authentication)