ARTESIS PLANTIJN
HOGESCHOOL ANTWERPEN

Projectverslag

Aan: Maarten Luyts & Tim Dams

Van: Michael Marivoet & Ward

Datum: 15/01/2016

Pagina's:

RFID cloner

Information Security

Inhoud

1 Uitleg Project

Wij hebben gekozen voor het project RFID Stealer. We vonden dit beiden een boeiend project aangezien we de voorbije jaren weinig hebben gewerkt met de beveiliging en de toegankelijkheid van bepaalde gevoelige data.

Dit project leek ons op het eerste zicht een interessant project omdat we op die manier konden laten zien hoe gemakkelijk het is om gevoelige data te pakken te krijgen zonder dat de gebruiker hiervan op de hoogte kan zijn.

Het werken met een Raspberry PI is in dat opzicht eveneens een openbaring voor ons. Ervoor werkten we vooral met Arduino's en VirtualBox om een programma te kunnen schrijven waarmee we informatie konden halen van derden.

Wat we zouden bekomen na het project leek me ook in dat opzicht een boeiend gegeven. Het verkrijgen van toegang tot een bepaalde module (sorteerstraat, klaslokaal, …) leek ons opwindend.

2 Belangrijke scripts

We zijn op verschillende manieren begonnen aan het project. Terwijl de ene persoon bezig was met de werking van een Raspberry PI uit te dokteren, was de andere bezig met het schrijven van een programma met Arduino om op die manier de gegevens die op een NFC kaart te verkrijgen en te kopiëren.

Nadat we besloten om zelf onze componenten aan te kopen, moesten we een aantal weken wachten op deze componenten. We konden wel al tutorials opzoeken die het mogelijk maakten om onze opdracht tot een goed einde te brengen.

De zaken die we zelf hebben aangekocht zijn RFID readers, lege kopieerkaarten en sleutelhangers die eigenlijk hetzelfde doen als de RFID readers. We hebben ook nog een aantal zaken gebruikt die we konden krijgen van de leerkracht. Dit waren Long-Range USB adapter voor de verbinding met Wifi, een Raspberry PI en de nodige kabels om alles met elkaar te kunnen verbinden zoals de verbinding tussen de Raspberry PI en een monitor om de Raspberry PI te kunnen configureren.

In het begin zochten we vooral naar een gemakkelijke manier om data te bekomen van een NFC kaart. Toen zochten we vooral naar manieren om dit te doen via de Raspberry PI, nadien wouden we ook eens kijken of er nog een andere manier was om dit te doen met een Arduino.

Bij punt 5.4 staat uitgelegd de uitleg waarom we uiteindelijk zijn overgeschakeld van Raspberry PI naar Arduino.

Dit bleek zo te zijn, er zijn al gemakkelijkere programma's geschreven voor Arduino die eigenlijk hetzelfde werk opknappen als een Raspberry PI. We besloten dan ook om te stoppen met zoeken naar code die we konden gebruiken voor de Raspberry PI

en ons te focussen op de code die gebruik maakt van een Arduino. Op die manier heeft ons dat veel tijd en vooral veel opzoekwerk uitgespaard.

Het programma dat voor Arduino geschreven is, is gevonden via een tutorial die allereerst een programma uitlegt waarmee je de info die op een NFC kaart staat, kan bekomen in de Seriële monitor.

Het tweede programma beschrijft een manier om enerzijds informatie te lezen en op te slaan in het geheugen van de Arduino en anderzijds om die gegevens ook te schrijven naar een nieuwe en lege NFC kaart.

Op die manier kopiëren we eigenlijk de data die op de NFC kaart staat.

Het probleem met dit programma is dat er zowel voldoende statisch geheugen als dynamische geheugen moet zijn om de data die staat op een NFC kaart te kunnen opslaan in het dynamische geheugen van de Arduino. We konden dit oplossen door een Arduino te nemen die een groter geheugen heeft dan diegene die normaal gezien op school beschikbaar zijn.

3 Werkwijze

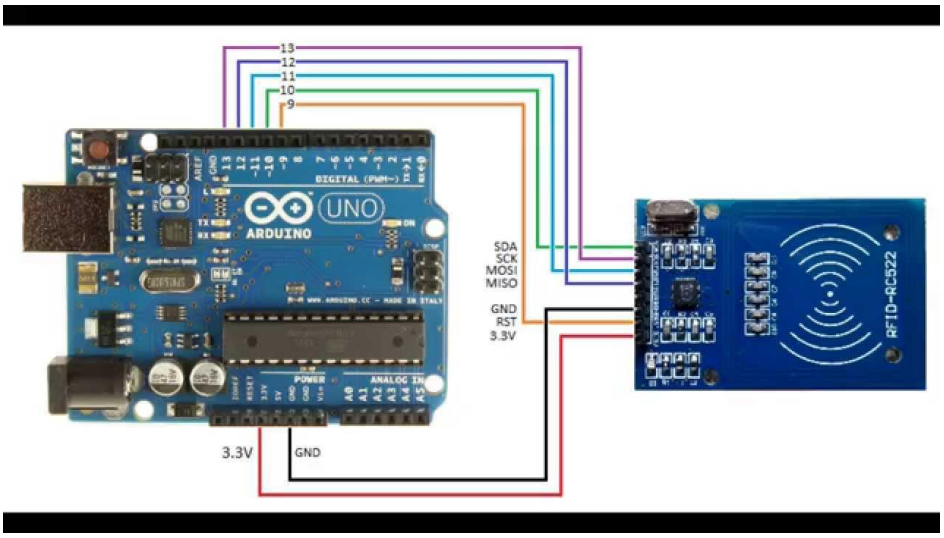
3.1 Verbinden van de RFID reader met de Arduino

Het verbinden van een RFID reader met een Arduino verschilt van Arduino tot Arduino.

Het verschilt in die mate doordat de MOSI, MISO en SCK verschilt van bordje tot bordje.

Signaal	RFID Reader Pin	Arduino Uno Pin	Arduino Mega Pin	Arduino Nano v3 Pin	Arduino Leonardo /Micro Pin	Arduino Pro Micro Pin
RST/ Reset	RST	9	5	D9	RESET / ICSP-5	RST
SPI SS	SDA (SS)	10	53	D10	10	10
SPI MOSI	MOSI	11 / ICSP-4	51	D11	ICSP-4	16
SPI MISO	MISO	12 / ICSP-1	50	D12	ICSP-1	14
SPI SCK	SCK	13 / ICSP-3	52	D13	ICSP-3	15

Bij een Arduino Uno geeft dit dan deze verbinding met de RFID reader.



Wat er echter niet verandert bij de verschillende Arduino's zijn de plaats waar de voeding(3,3 V), de grond en Slave Select (SS) worden aangesloten met de RFID reader.

3.2 Schrijven van code

Deze taak spreekt voor zich, we proberen zo veel mogelijk op te steken van de tutorials en library's om zo een voor ons zo best begrijpbare code te schrijven.

3.3 Verifiëren van de code

We passen de code aan als er tijdens het compileren fouten zijn gevonden door de compiler. Indien nodig zoeken we op het internet naar veel gemaakte fouten om ons op die manier te helpen.

3.4 Uploaden van de code

We laden onze code op het bordje en testen het programma uit. Indien er iets fout loopt, dan passen we de code aan of zoeken we de fout op het internet op.

De meest gemaakte fout in deze context is het onjuist kiezen van het soort van bordje en COM poort. Een van ons had het probleem dat er geen COM poort kon geselecteerd worden. Dit werd verholpen door een ander bordje te kiezen.

Een andere fout was dat het dynamische geheugen van de poort te klein was om de lokale variabelen te kunnen opslaan. Hier werkt oftewel het veranderen van chip maar wel hetzelfde bordje blijven behouden oftewel het veranderen van bordje

Als alles goed is verlopen, testen we of het programma ook effectief werkt. We hebben ons programma bij voorbeeld getest met de lege NFC kaart en de sleutelhanger die we hebben gekregen bij de RFID reader.

4 Gebruikte libraries

We hebben voornamelijk de SPI library gebruikt om de communicatie tussen de RFID reader / writer met de Arduino te vergemakkelijken. Net zoals bij de eerste praktijkles van Internet Of Things maar in dit geval om 1 Arduino en 1 RFID reader/ writer te laten communiceren in plaats van 2 Arduino's met elkaar te laten communiceren.

Daarnaast hebben we ook nog de MRFC22 library gebruikt om de RFID reader / writer aan te kunnen spreken en functies die oftewel in de library staat of een zelf geschreven functie uit te voeren. Dit is de library van de component die we hebben aangekocht.

5 Verloop

Door Michael Marivoet.

5.1 Materiaal

Ik heb direct nadat ik de opdracht had gekozen, op zoek gegaan waar ik een goedkope RFID lezer/schrijver kon kopen en wat type dat deze het best kon zijn. Na wat Googlen en het vinden van meerdere bronnen/tutorials over de Raspberry Pi en de RC522 module hebben we gekozen om deze te kopen. We hebben de module aangekocht via DealExtreme (link bij bronnen), deze was na ongeveer 9 dagen geleverd. De module kwam met een RFID sleutelhanger en kaart. Deze zijn beide MILFARE classic kaarten van 1K. Over een eigen Raspberry Pi model B beschikte ik al. De RC522 module gebruikt SPI om te communiceren met de Raspberry Pi en heeft 3,3V nodig om te kunnen werken.

Omdat ik in een later stadium van ons project ben overgeschakeld van een Raspberry Pi naar Arduino, ben ik begonnen met het gebruiken van men eigen Arduino Mega.

5.2 Onderzoek

Allereerst zijn we begonnen met te zoeken naar informatie, tutorials, scripties, etc. die over het lezen, schrijven, kopiëren, etc. van RFID tags gaan. We hebben veel geleerd over de MIFARE classic RFID tag dankzij de scriptie van Wee Hon Tan. De tutorials van MakeCourse legt uitbundig uit hoe RFID werkt en hoe de EEPROM eruit ziet.

Als code voorbeelden hebben we code gebruikt van MakeCourse en RFID/Arduino: copy a card with know keys. Voor informatie hierover en een link naar de bronnen, zie bij punt 6, bronnen.

Omdat we initieel voor Raspberry Pi gingen, hadden we eerst bronnen over RFID en Raspberry Pi.

5.3 MILFARE classic

We zijn ons gaan focussen op de MILFARE classic kaarten, omdat we deze bij onze RFID module zat en de school zijn studentenkaarten ook van dit type zijn. We hebben later ontdekt toen we de studentenkaarten aan het uitlezen waren dat die van het type 4K zijn en dus meer geheugen hebben. De RFID chip gebruikt EEPROM (Electrically Erasable Programmable Read-Only Memory) om data op te slaan.

Op Wikipedia en in andere bronnen hebben we ontdekt dat de simpele versleuteling van de MILFARE classic al lang omzeild was. Toch werden deze kaarten nog veel gebruikt, omdat deze zeer goedkoop zijn in aanschaf.

De classic 1K kaart beschikt over 16 sectoren met ieders 4 blocks van 16 Bytes (16*4*16 = 1024 = 1KByte). Dit wil zeggen dat er in totaal 64 blocks zijn, van elke sector zijn 4 blocks bevat er eentje de 2 beveiligingssleutels (key A & B). Deze kan uniek zijn voor elke sector en is standaard wanneer deze van de fabrikant komt 255 (DEC) of 1111 1111 (BIN) voor elke sector. De security block ziet er als volgt uit, 6 bytes voor key A, hierna 4 bytes voor access bits en als laatste nog is 6 bytes voor key B. De beveiliging vereist maar 1 key om te werken hierdoor is de 2de dus optioneel, maar verhoogt de beveiliging wel.

De 4K variant, gebruikt door de school voor de studentenkaarten, heeft het zelfde beveiligingsprincipe, maar met 4x zoveel geheugen en de indeling is hier ook iets anders. Deze versie heeft in totaal 40 sectoren. Hier bestaan de eerste 32 sectoren ook uit 4 blocks van 16 Bytes zoals de 1K versie, maar de laatste acht sectoren bestaan uit 16 blocks van 16 Bytes.

Block 0 van sector 1 bevat de UID (4 bytes), BCC (1 byte) en fabrikant zijn gegevens (10 bytes). De UID kan gebruikt worden om een unieke key B te genereren.

De RFID beschikt over 6 memory operations, namelijk read, write, increment, decrement, transfer, restore.

Operation	**Description**	**Valid for Block Type**
Read	Reads a single memory block	read/write, value and sector trailer
Write	Writes a single memory block	read/write, value and sector trailer
Increment	Increments the value and stores the result in the internal data register	value
Decrement	Decrements the value and stores the result in the internal data register	value
Transfer	Transfers contents of internal data register to a block	value
Restore	Reads contents of a block into the internal data register	value

Na het uitlezen van mijn studentenkaart ben ik erachter gekomen dat deze leeg is en nog altijd de factory key heeft.

5.4 Van Raspberry PI naar Arduino

Oorspronkelijk zijn we begonnen met Raspberry Pi, maar we hadden hier al vanaf het begin problemen mee. Ik heb in totaal 3x een nieuwe (verse) versie van Raspbian moeten installeren, doordat deze corrupt geraakte en in een bootloop bleven hangen. Hierdoor ben ik enkele malen mijn python scripts kwijt geraakt, door niet op tijd en back-up te voorzien.

Nadat ik hoorde dat we ook met een Arduino mochten werken, ben ik hiermee opnieuw begonnen. Na wat zoeken vond ik een tutorial die uitlegt hoe RFID communicatie werkt op MakeCourse en een ander persoon zijn project rondom het klonen van RFID tags me gekende sleutels. De maakten beide gebruik van de MFRC522 library door Miguel Balboa (zie bronnen).

We hebben we ontdekt dat de Arduino Uno bijvoorbeeld niet genoeg geheugen (SRAM) heeft om een tag te kunnen klonen. Gelukkig beschik ik over een Arduino Mega die 4x zoveel geheugen heeft (8K i.p.v. 2K).

5.5 Dumpen van RFID tag informatie

Met behulp van een tutorial hebben we de RFID data kunnen dumpen via de seriële poort, zodat we deze kunnen analyseren. Het is zo dat we geleerd hebben dat de studentenkaart leeg is.

5.6 Crypto

MIFARE classic gebruikt Crypto1 dat eigendom is van NXP Semiconducters, het algoritme is geïmplementeerd als hardware on-chip voor snelle codering van informatie.

6 Eindstatus project

Momenteel kunnen we buiten het lezen van en dumpen van de RFID kaart niet eigenlijk niets, dit komt mede doordaat dat we de eerste 4 weken gewerkt hebben met de Raspberry Pi en dit tot niets heeft geleid.

Momenteel was ik nog bezig met het kunnen lezen, opslaan en dan schrijven van RFID tags waarvan we de keys kenden, maar door problemen met de library heb ik dit niet in orde gekregen.

7 Bronnen

7.7 *Practical Attacks on the MIFARE classic by Wee Hon Tan*

7.8 *MakeCourse: using MF522 RFID reader with Arduino*

7.9 *RFID/Arduino: copy a card with known keys*

7.10 *Arduino RFID library for MFRC522 – Miguel Balboa*

7.11 *Link RC522 RFID lezer/schrijver dealextreme.com*

7.12 *MIFARE RFID – Wikipedia*

7.13 *RFID - Wikipedia*

8 Link naar GitHub Repository

GitHub Repo