# Labo opdracht 3: NFC – BLE – Wireless Tag Scanner

## Inleiding

### Omschrijving opdracht

De idee van onze labo opdracht is om een draadloze NFC scanner te maken. Deze scanner kan NFC-tags uitlezen in bijvoorbeeld warenhuizen waarbij men kan zelf scannen tijdens het winkelen. De klant krijgt op zijn of haar smartphone of tablet een winkellijst te zien met de gescande producten.

### Taakverdeling

Korneel: App, concept

Emiel: Micro-zijde, concept

## Microcontroller Code

### Opzet

Voor de micro controller zijde hebben we beslist om eerst een 4 tal uur te proberen de 2 voorbeeldcodes ( BLE en NFC) samen te voegen in 1 enkel project . We hebben meteen een maximum grens qua tijd gezet omdat we wisten dat dit een ambitieuze onderneming zou zijn maar met een mooist mogelijke resultaat.

Een link naar de Github van deze pogingen vind u hier:

<https://github.com/Fishezzz/Micro-Project-BLE-NFC>

Info:

3.1: NFC en BLE vanuit STM Cube vanaf 0 opbouwen ( kon niet builden)

3.2 eerst Standalone poging NFc

3.3 Laatste poging om NFC en BLe samen te voegen vanuit de examples.

3.4 NFC en BLE in twee apparte projecten met Uart als communicatie.

3.5 kopie van 3.4 + DMA of interrupt 🡪 geraakt niet voorbij MS\_BlueNRG\_MS\_Init();

In een verdere fase hebben we dan overgeschakeld naar een andere tactiek waarbij we de twee projecten apart met elkaar de tag laten doorgeven via UArt.

### NFC -Board

We verwijderen enkele boodschappen op de uart in mian.c

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

En:

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

En verkrijgen bij het aanleggen van de kaart enkel de gewenste ID binnen van het product:



### BLE-Board

Hier veranderen we in app\_x-cube-ble1.c het server adress naar het gekozen adres:

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

Alsook in sample\_service.c:

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

Eventueel kan ook de naam aangepast worden van het BLE apparaat maar daar maken we in feite geen gebruik van in de app:

Afbeelding met schermafbeelding

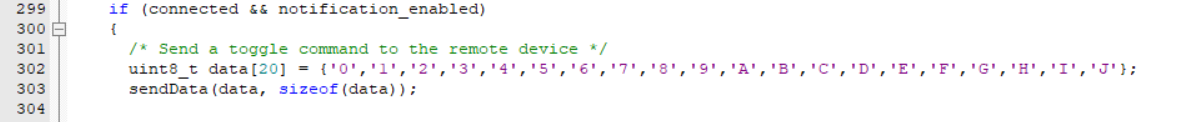
Automatisch gegenereerde beschrijving

De volgende UUID waren van belang om de jusite subscription te maken vanaf de app:

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

Bij wijze van test en proof of concept ( wanneer de uart zou werken tussen Ble en NFC) zou hier de juiste date moeten ingegeven worden. ( in app\_x-cube-ble1.c)



En vervangen dit door :

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

## Android-app code

### Opzet

We hebben de app gemaakt via MIT-app inventor. Om zich eigen te maken aan de manier van werken met de UI en de logica van app inventor heb ik eerst volgend voorbeeld gevolg:

<http://iot.appinventor.mit.edu/assets/tutorials/MIT_App_Inventor_Basic_Connection.pdf>

In een volgende stap heb ik een automatische connectie proberen leggen met het bluetooth apparaat zodat niet telkens uit een lijst moest geselecteerd worden maar op een bepaalde Mac-adres verbind.

### Oefenapp 1:Miauw

Deze simpele oefening was de eerste app met app-inventor die ik maakte:

### Afbeelding met schermafbeelding Automatisch gegenereerde beschrijving

### OefenApp 2: Bleutoot oplijsten en connecteren

#### GUI:

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

#### Code:

Wanneer het scherm geladen is wordt de shop knop zichtbaar.

Wanneer er ok op knop btnScan geklikt wordt het scannen opgestart. En de status geüpdatet en de list geactiveerd.

Wanneer op de stopknop wordt gedrukt stopt het scannen.

Afbeelding met schermafbeelding, teken

Automatisch gegenereerde beschrijving

Waneer een bleutooth apparaat werd gevonden wordt deze toegevoegd aan de listview. Hier is ook een eerste aanzet om automatisch te connecteren ingewerkt.

Verder Kan met ook connecteren door een item uit de listview te selecteren en op connect te drukken.

Ten slotte wordt bij een fout tijdens het connecteren deze opgevangen en getoond op het berichtvenster:

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

#### Resultaat:

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

### AutoConnect App

Dit is de definitieve app die automatisch connecteert op een welbepaalde NFC-scanner. En bij elk gescand item, de productcode moeten ontvangen.

#### GUI

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

#### Code

Eerst worden enkele globale variabelen aangemaakt: Het mac adres van de bLE, een lijst om de mogelijke Ble apparaten in op te slaan en de Service en Char \_UUID om de correct te kunnen subscriben op de microcontroller.

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

Wanneer het scherm geladen wordt begint het scannen automatisch:

Afbeelding met lucht, schermafbeelding

Automatisch gegenereerde beschrijving

Vervolgens wordt er telkens wanneer er een apparaat word gevonden deze aan een (onzichtbare lijst ) toegevoegd. Vervolgens filteren we het max adres er uit en steken de resultaten in een nieuwe lijst.

We proberen te connecteren op 69:69:69:69:69:69 en zoeken vervolgens in onze lijst met resultaten of deze aanwezig is, zo ja, verbinden we.

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

Vervolgens subscriben we ons op het juiste service en char\_UUId wanneer de BLE connected is.

En passen de boodschappen aan.

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

Wanneer we een string verkrijgen via onze subscription, voegen we deze toe aan de boodschappenlijst en tonen deze als laatst gescand.

Afbeelding met schermafbeelding, binnen

Automatisch gegenereerde beschrijving

Tenslotte proberen geven we een boosdchap wanneer de scanner zou deconnecteren.

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

#### Resultaat:

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

Op de moment van het opmaken van het verslag kon de laatste stap spijtig genoeg niet uitgevoerd worden. Tijdens het labo heb ik echter een positieve test gehad waarbij na het aanmaken van een geldige subscription de gescande tag in de lijst kwam.

Dit heb ik enkel kunnen simuleren door het drukken op de blauwe knop op de FR401 en zo ene bericht te verkrijgen.

## Besluit

In het begin van het project was het even onduidelijkheid in verband met de Mac-adressen van de BLE apparaten. We gingen er vanuit dat elk Mac-adres uniek was. Dit was echter niet zo . Doordat elke leerling startte van hetzelfde voorbeeldproject stonden deze allemaal op hetzelfde adres om te beginnen. En was het onmogelijk om je eigen apparaat te vinden . Nadat we ons eigen adres programmeerden was dit snel opgelost.

Binnen de voorbeelden van NFc en BLE vonden we vrij snel onze weg en konden we vrij snek zeggen wat er waar moest aangepast worden om tot een resultaat te komen .

De grootste moeilijkheid was om de uart communicatie tussen de twee borden in orde te krijgen. Waarin we uiteindelijk niet in geslaagd zijn, desondanks menige pogingen vanuit verschillende insteken.

We hadden ons misschien eerst de UArt proberen laten werken en dan pas de twee projecten proberen samenvoegen.

Tijdens het programmeren van de app merkte ik wel veel onstabiliteit in het bereik van de bluetooth . Het lukte vaak niet om te verbinden en dan plots wel, zonder dat ik daar een goede reden voor kon vinden.

Het werken met de app-inventor was zeer strait forward en een leuk intermezzo op het puur microcontroller gebeuren, waarbij het resultaat soms haaks staat op de gepresteerde uren en inspanning.

De bestanden van het project zal u terugviden in de git repository onder Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

Op onze Github repository:

<https://github.com/Fishezzz/Micro-Project-BLE-NFC>