

Aan de slag met

USB4NEMO

Voor eerste jaars nemo studenten

Met programmeer voorbeelden

Auteur: O. Chougna
USB4NEMO Team

Inhoudsopgave

Inhoudsopgave	2
Inleiding.....	3
Voordat je begint.....	4
De Microcontroller programmeren	4
Windows PC configureren.....	4
Usb4nemo apparaat testen	5
De richting van een pin.....	5
Waardes op een pin zetten	5
Poll AD knoppen.....	6
Aan de slag met programmeren	6
Programmeer omgeving instellen.....	6
Usb4nemo klassen.....	6
Usb4nemoDevice.class	7
USBController.class	7
Port.class	7
ADConverter.class.....	7
DeviceNotFoundException.class.....	7
Usb4NemoProtocolPacket.class.....	7
Voorbeeld programma's	7
Voorbeeld programma 1: Leds aanzetten	7
Voorbeeld programma 2: Een poort inlezen	8
Voorbeeld programma 3: Een ADConversie uitvoeren.....	9

Inleiding

Dit document is bedoeld voor eerste jaars studenten technische informatica die het project Nemo doen. Usb4nemo is een hulpmiddel voor het Nemo project waarbij gebruik wordt gemaakt van de USB interface. Om met dit document te beginnen heb je de hardware al af. Zie het document hardware van Usb4nemo voor meer informatie over de hardware. (Solderen en etsen) Dit document bevat ook programmeer voorbeelden en hoe je de hardware kan testen op functionaliteit.

Voordat je begint

Voordat je begint heb je de hardware al af. Ook heb je al een Pic18F4550 microcontroller en heb je deze geprogrammeerd met het usb4nemo.hex bestand (Zie volgende hoofdstuk)
Het volgende heb je dus nodig:

- Usb4nemo.hex
- Usb4nemo.dll
- De JavaAPI samen met de Javadocs
- Test programma, usb4nemo.jar
- Windows XP/2003 PC met Java geïnstalleerd. (JDK en JRE)

De Microcontroller programmeren

Het eerste wat je moet doen is de microcontroller programmeren. Het type microcontroller wat je nodig hebt is de **PIC18F4550**. De software hoeft je niet zelf te schrijven. Deze microcontroller kun je bestellen bij Microchip. (www.microchip.com)
Ook kun je proberen de microcontroller als sample te bestellen bij microchip. Dit is gratis.

Als je de microcontroller in je bezit hebt dan moet je er firmware op zetten. Dit is het usb4nemo.hex bestand. Het beste is om een Galep of Leaper te gebruiken, deze zijn allebei op school aanwezig. Dit zijn universele programmers waarmee je je PIC18F4550 kunt programmeren. De software kun je via Internet downloaden. (Even Googlen naar galep)
Zodra usb4nemo.hex op je microcontroller staat dan ga je je windows PC configureren voor gebruik met usb4nemo.

Windows PC configureren

Usb4nemo is alleen te gebruiken met Windows XP of Windows 2003, helaas is het niet mogelijk om Linux/Unix te gebruiken. Usb4nemo gebruikt een bibliotheek om met de USB bus in windows te communiceren. Deze bibliotheek is usb4nemo.dll, dit bestand moet je op de volgende plek neerzetten:

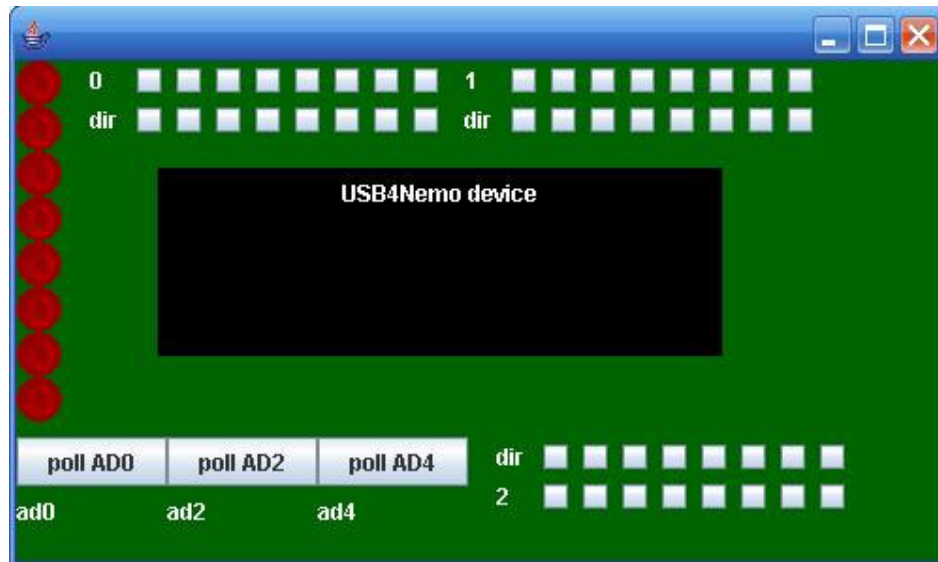
%Systemroot%/System32/

%Systemroot% is de root van je Windows installatie, meestal is dit C:\Windows.

Zodra je usb4nemo.dll in je system32 map hebt gezet kun je jouw usb4nemo apparaat gaan testen met het usb4nemo programma, usb4nemo.jar.

Usb4nemo apparaat testen

Als je een usb4nemo apparaat goed hebt gesoldeerd en geëetst, dan zou het goed moeten gaan werken met het test programma usb4nemo.jar. Dit is een grafisch programma waarmee je het apparaat kan testen. Zo kun je op de leds klikken en kijken of al je leds aan en uit kunnen. Ook kun je kijken of de poorten goed functioneren door ze actief te maken met een vinkje en te kijken, met bijvoorbeeld een multimeter, of er 5 Volt op de poort staat.



Usb4nemo.jar test programma

De rode cirkels stellen je leds voor, de vierkantjes stellen je IO pinnen voor, elk pinnetje heeft 2 aanvink vakjes, eentje voor de richting van de pin (dir) en eentje voor de waarde die je op de pin zet. Het usb4nemo apparaat heeft 3 pin poorten; poort0, poort1 en poort2. Verder kun je met dit programma de ADConverter kanalen testen.

De richting van een pin

Met de richting van een pin bedoelen we of de pin een input pin is of een output pin. Een output pin kun je uit sturen. Je kan daar dus een 1 of een 0 opzetten, respectievelijk 5 volt of 0 volt. Een input pin kun je inlezen, dus je kan kijken of er 5 volt van buitenaf op is gezet of 0 volt. De vinkjes kun je neerzetten bij de dir vinkvakjes.

Aangevinkt: Pin is output.

Niet aangevinkt: Pin is input

Waardes op een pin zetten

Op elke pin kun je een waarde neerzetten, een 1 of een 0. Respectievelijk is dit 5 Volt of 0 Volt. Zet je een pin bijvoorbeeld op 1 dan stuurt de pin een 5 volt waarde uit.

De bovenste aanvinkvakjes kun je hiervoor gebruiken:

Aangevinkt: Er staat een 1 op de pin (5 Volt)
Niet aangevinkt: Er staat een 0 op de pin. (0 Volt)

LET OP:
Denk goed na bij wat je doet. Vermijd kortsluiting!!

Poll AD knoppen

Op het usb4nemo apparaat zitten 5 Analooog naar Digitaal converter kanalen. Met een ADConverter kun je analoge waarden omzetten naar digitale waarden. Op elk kanaal kun je een waarde tussen de 0 en 5 volt zetten, als je daarna op de poll ADX knop drukt, waarbij X overeenkomt met het kanaal dat je gebruikt, dan krijg je de waarde onder de knop te zien.

Aan de slag met programmeren

Als je usb4nemo apparaat helemaal goed functioneert dan kun je aan de slag met programmeren in Java.

Programmeer omgeving instellen

Voordat je in java je usb4nemo applicatie gaat programmeren moet je ervoor zorgen dat je programmeer omgeving helemaal goed ingesteld staat. Je kan Eclipse, JCreator of BlueJ gebruiken. De usb4nemo JavaAPI bevat alle klassen die je nodig hebt, zorg ervoor dat je deze klassen aan je Classpath toevoegt voordat je begint.
Ook heb je de javadocs nodig. Dit is de documentatie die bij de usb4nemo klassen hoort. (HTML pagina's) Deze staan in de doc map.

Usb4nemo klassen

De usb4nemo klassen die je nodig hebt om je eigen applicatie te bouwen zijn:

Usb4nemoDevice.class
ADConverter.class
Port.class
USBController.class
DeviceNotFoundException.class
Usb4NemoProtocolPacket.class

Hieronder volgt een korte bespreking

Usb4nemoDevice.class

Deze klasse vormt de basis van je klasse, de klasse stelt jouw usb4nemo apparaat voor. Een Usb4nemoDevice klasse instantie bevat de volgende onderdelen:

- 5 ADConverter instanties, ad0 tot en met ad4
- 3 Port instanties port0 tot en met port2.
 - Op port0 zitten de Leds en IO pinnen
 - Port1 bevat 8 pinnen
 - Port2 bevat 8 pinnen
- Een USBController instantie.

USBController.class

Deze klasse verzorgt de communicatie tussen jouw usb4nemo device en de pc.

Port.class

Deze klasse stelt een poort voor, een poort bevat 8 pinnetjes. Met een poort kun je van alles doen, je kan de richting van een pin veranderen of bijvoorbeeld een waarde op een bepaalde pin zetten. Kijk in de javadocs voor meer informatie.

ADConverter.class

Deze klasse stelt een ADConverter kanaal voor. De usb4nemodevice klasse bevat standaard al 5 ADConverter kanalen. Kijk in de javadocs wat je zoal met een ADConverter kanaal kan doen.

DeviceNotFoundException.class

Deze klasse is een Java Exception, deze exception wordt opgegooid wanneer je een usb4nemodevice probeert te instantieren terwijl het usb4nemo apparaat niet is aangesloten.

Usb4NemoProtocolPacket.class

Deze klasse is niet van belang.

Voorbeeld programma's

In dit paragraaf geven we een aantal programmeer voorbeelden voor usb4nemo.

Voorbeeld programma 1: Leds aanzetten

Om de Leds aan te zetten volg je de volgende stappen:

1. Maak een usb4nemodevice object aan.
2. Pak van het usb4nemo object poort 0

3. Zet de pinnen van de poort in de juiste richting.
4. Activeer alle pinnen

1. Maak een usb4nemodevice object aan.

```
USB4NemoDevice usb4nemodev = new USB4NemoDevice();
```

2. Pak van het usb4nemo object poort 0

```
Port Leds = usb4nemodev.getPort0();
```

3. Zet de pinnen van de poort in de juiste richting

```
Leds.setPortPinDirections((byte) 0);           //Alle pinnetjes als output
```

4. Activeer alle pinnen

```
Leds.setPortPinValues((byte) 255);           //Alles hoog, dus alle leds aan
```

Voorbeeld programma 2: Een poort inlezen

Een usb4nemo device heeft 3 poorten, poort0, poort1 en poort2. Om een poort in te lezen dien je de volgende stappen te volgen.

1. Maak een usb4nemodevice object aan
2. Pak de gewenste poort
3. Stel de poort in als input, zodat je de poort kan inlezen
4. Lees de waarde van de poort uit.

1. Maak een usb4nemodevice object aan.

```
USB4NemoDevice usb4nemodev = new USB4NemoDevice();
```

2. Pak van het usb4nemo object een poort (0,1,2)

```
Port poort = usb4nemodev.getPortX();           //ipv X, 0,1 of 2 gebruiken
```

3. Stel de poort in als input

```
Poort.setPortPinDirections((byte) 255)           //Poort instellen als input
```

4. Lees de waarde van de poort uit

```
Byte waarde = poort.getPortPinValues();
```


Voorbeeld programma 3: Een ADConversie uitvoeren

Het usb4nemo apparaat heeft 5 ADConverter kanalen, elk kanaal is gekoppeld aan een pin. Een ADConversie kun je als volgt doen.

1. Maak een usb4nemodevice object aan.
2. Pak het gewenste AD Converter kanaal.
3. Doe een ADConversie.
4. Lees het resultaat van de ADConversie uit.

1. Maak een usb4nemodevice object aan

```
USB4NemoDevice usb4nemodev = new USB4NemoDevice();
```

2. Pak het gewenste AD Converter kanaal.

```
ADConverter kanaal2 = usb4nemodev.getAD2();
```

3. Doe een ADConversie

```
Kanaal2.poll();
```

4. Lees het resultaat van de ADConversie uit

```
Double resultaat = kanaal2.getValue();
```

Het resultaat is een waarde tussen 0 en 5 volt.