

Labo datacommunicatie : Seriële communicatie

Doelstelling

- Aanspreken van de seriële poort vanuit .NET
- Data uitwisselen via seriële communicatie.
- Implementatie van een protocol

Benodigheden

- Null-modem kabel : Kabel voor seriële communicatie van PC tot PC
- USB/RS232 convertor
- Virtual serial port kit.

Inleiding

Hoewel de plaats van de seriële poort steeds meer verdwijnt in het dagdagelijkse huis-, tuin en keuken gebruik van de PC, is de nood aan seriële communicatie (en aanverwante technologieën) in de industrie nog steeds erg hoog. Vaak is het ook zo dat de fysieke verbinding vervangen is door b.v. USB maar de logische implementatie gebeurt dan door een virtuele seriële poort.

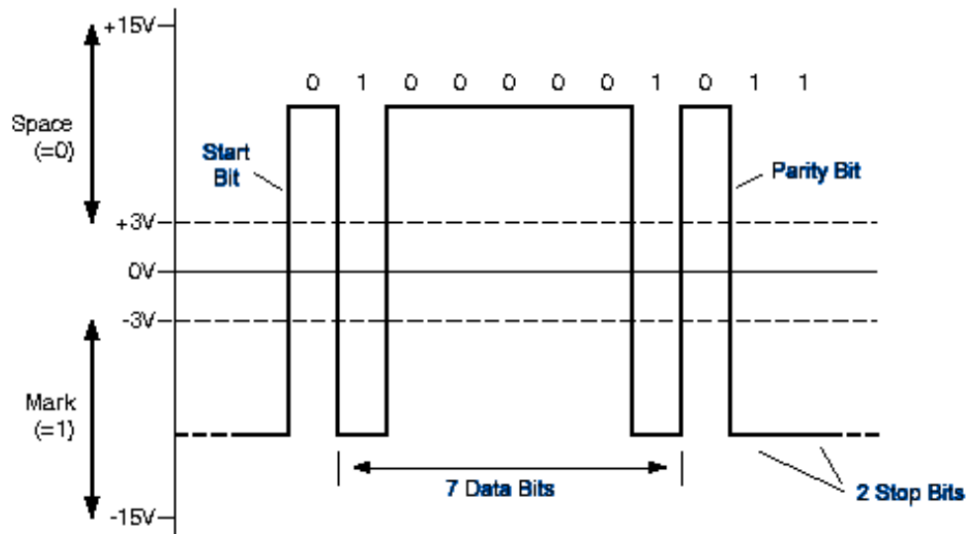
De seriële poort

De seriële poort op de pc is wat men noemt een asynchrone seriële interface d.w.z. dat er geen synchronisatie- of kloksignaal op de poort aanwezig is, zodat er met elk willekeurig interval tekens kunnen worden verzonden. Het woord 'asynchroon' betekent dat de informatie niet op vooraf afgesproken momenten wordt verzonden. Het verzenden van de data kan op elk willekeurig moment starten en het is daarbij de taak van de ontvanger om te detecteren wanneer een boodschap start en eindigt. Met het woord serieel bedoelen we dat de informatie bit per bit wordt verzonden.

Elk teken bevindt zich tussen een start- en stopsignaal. Aan elk teken gaat één 0-bit vooraf, de startbit, om het ontvangende systeem te vertellen dat de volgende zeven of acht bits een databyte vormen. Het teken wordt telkens afgesloten door één of twee stopbits. Aan de kant van de ontvanger worden de tekens aan het start- en stopsignaal herkend.

Databits worden met een vastgelegde frequentie verzonden, de baud rate. Zowel de zender als de ontvanger moeten geprogrammeerd zijn voor gebruik van dezelfde bitfrequentie. Nadat de eerste bit ontvangen is berekent de ontvanger op welke momenten de volgende databits kunnen worden verwacht. Hoe lager de baudrate, hoe minder snel maar hoe groter de maximale kabellengte.

Merk op dat een positieve spanning een lage bit betekent, de negatieve spanning is een hoge bit.



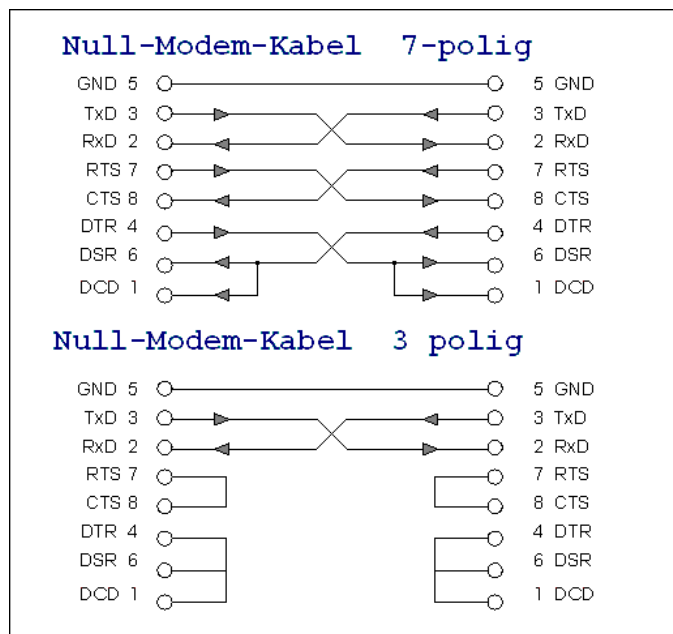
De flowcontrol kan op drie manieren gebeuren.

- Geen (gaan wij vandaag toepassen)
- Softwarematig (via het sturen van XON/XOFF bytes)
- Hardwarematig.(RTS/CTS handshaking)

D-Type-25 Pin No.	D-Type-9 Pin No.	Abbreviation	Full Name
Pin 2	Pin 3	TD	Transmit Data
Pin 3	Pin 2	RD	Receive Data
Pin 4	Pin 7	RTS	Request To Send
Pin 5	Pin 8	CTS	Clear To Send
Pin 6	Pin 6	DSR	Data Set Ready
Pin 7	Pin 5	SG	Signal Ground
Pin 8	Pin 1	CD	Carrier Detect
Pin 20	Pin 4	DTR	Data Terminal Ready
Pin 22	Pin 9	RI	Ring Indicator

Table 1 : D Type 9 Pin and D Type 25 Pin Connectors

Null modemkabel



Seriële Communicatie in .NET

Vroeger werd de seriële poort aangesproken via een ActiveX control mscomm. In .NET wordt er echter geen gebruik meer gemaakt van ActiveX en daarom heeft Microsoft een speciale klasse voorzien om seriële communicatie te verzorgen in .NET.

Het is zelfs zo dat Microsoft vanaf Visual Studio 2005 terug een control introduceert om aan seriële communicatie te doen. Deze trend illustreert het belang van seriële communicatie.

We kunnen in de code een instantie aanmaken van de klasse SerialPort.

Bij de properties, vullen we de verschillende parameters in. Zorg ervoor dat zender en ontvanger dezelfde parameters meekrijgen. (baudrate, stopbits,...) Vergeet niet om de flowcontrol in te stellen. Vul bij readtimeout 10ms in, dit betekent dat de read methode 10ms wacht op data.

De belangrijkste methodes in de class RS232 zijn:

- **Open()** : Deze methode dient om de communicatie met een seriële poort te openen.
- **Close()**: De connectie met de seriële poort sluiten
- **WriteLine()**: Deze methode laat u toe om een string naar de seriële poort te schrijven. Die wordt beëindigd met een CRLF.
- **ReadExisting()** Hiermee worden alle beschikbare ingelezen bytes gelezen, en als string returned.
- **Read(...)**: Deze methode probeert om een aantal bytes te lezen uit de ontvangbuffer. Deze bytes komen in een chararray terecht van 100 bytes groot. Als het niet tijdig lukt om 100 bytes in te lezen (timeout) dan is er een exception; Die moeten we opvangen. Toch zijn er een aantal bytes gelezen.
- **DataReceived event** : dit event wordt opgeroepen als er bytes binnenkomen.

Buffers

In elk com device is er een ingebouwde zend- en ontvang buffer. Deze buffer vangt bytes op die niet onmiddellijk door software gelezen worden.

Indien er data verstuurd wordt, bijvoorbeeld : "Dit is een test met de seriële poort.". Is er een stroom bytes die naar de andere pc vertrekt. Deze komt daar aan in de buffer. De grootte van de buffer is instelbaar in Apparaatbeheer/ Communicatiepoort/ geavanceerd/ . De toepassing die de compoort open heeft, kan deze bytes ontvangen. Als een toepassing elke seconde data uit de buffer leest is de kans groot dat de verstuurde string in één leesoperatie kan worden ingelezen. Dit is echter niet zeker. Het kan best zijn dat de leesbuffer wordt ingelezen vooraleer de volledige data is verstuurd. Eén seconde later wordt dan de rest van de bytes ingelezen. Onze "testzin" kan dus in meerdere keren ingelezen worden. Hou dus steeds rekening met het feit dat bij een leesoperatie slechts een aantal bytes zullen worden gelezen en niet het volledige bericht.

Andere instellingen

Het timerinterval zetten we op 200 ms.

Het volgende probleem kan zich voordoen: De gegevens worden sneller gelezen dan ze verstuurd worden. Als de string bijvoorbeeld in hyperterminal wordt ingetikt, duurt het enkele seconden vooraleer het intikken is beëindigd. Telkens een toets wordt ingetikt, wordt deze serieel verstuurd, en komt die ook aan in de buffer van de ontvangst pc. De leestoepassing kan bijvoorbeeld elke 100ms uit de buffer gaan lezen. Daar zal meestal niks of slechts één byte in aanwezig zijn. Het volledige bericht zal dus niet in één leesoperatie kunnen worden gelezen. Elke leesoperatie zal een deel van de string binnenlezen. Het is dus aan jullie leesalgoritme om die tekens terug aan elkaar te hangen tot een woord of een zin.

Protocol:

Om de toepassing hier te gaan programmeren, bedenk je een protocol. Een protocol is eigenlijk niet veel meer dan een set afspraken. Aangezien we bij seriële communicatie vaak tekstuele of binaire berichten gaan doorsturen, is het protocol niet meer dan een zelf samengestelde lijst van commando's en parameters die door zender en ontvanger worden begrepen. Bij elk commando hoort dikwijls ook een vooraf vastgelegde lijst van mogelijke antwoorden die kunnen worden teruggestuurd.

Dit heeft als doel dat de lees routine weet wanneer een commando is beëindigd, zodat dit commando kan verwerkt worden. De ontvanger stuurt bijvoorbeeld ACK door als het commando is herkend, en NACK als het commando niet is herkend. We zullen commando's intikken in hyperterminal. We kunnen bijvoorbeeld ledsAan# intikken, waarbij # betekent dat het commando is beëindigd. ledsUit#, enz... In plaats van # kunnen we CR(carriage return) of LF(line feed) gebruiken, Dit werkt handig met hyperterminal. CR en LF worden steeds gebruikt in tekstbestanden om een nieuwe lijn te beginnen. Zo moet je maar eens een tekstbestand openen met visual studio in binaire vorm(open with in de open file dialog). Dan zie je de cr en lf staan als 13 10(decimaal) of 0D 0A hexadecimaal

Speciale karakters in .NET

Soms kan het handig zijn als je een speciaal karakter zoals een enter of een tab kan doorsturen via een string. Je kan dit doen door gebruik te maken van Character literals.

[http://msdn.microsoft.com/en-us/library/aa691087\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/aa691087(v=vs.71).aspx)

\r : Carriage Return

\n : Line Feed

\r\n : Carriage Return / Line Feed = ENTER

\t: een tab

Praktisch

De meeste laptops beschikken niet over een seriële poort. Vandaar de volgende twee mogelijke oplossingen:

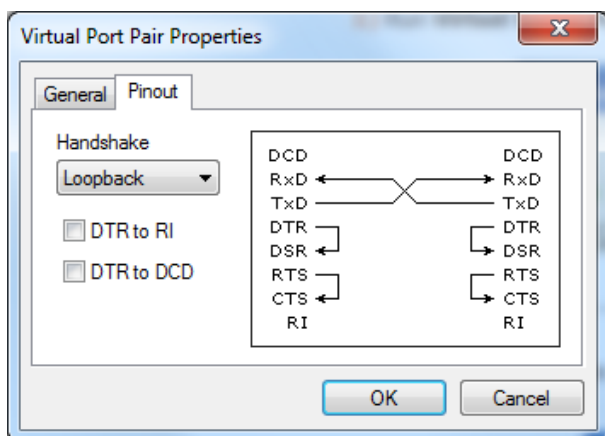
1. USB-RS232 met NULLMODEMKABEL

Er zijn 20 USB-RS232 kabels beschikbaar. Met deze adapters is het mogelijk om een seriële poort toe te voegen aan uw laptop. De driver staat bij dit document op de server. Let erop dat de compoort een nummer krijgt toegewezen dat kleiner is dan 5. Dit omdat er vroeger maximum 4 seriële poorten aanwezig konden zijn in een computer. Vele software werkt daardoor niet goed bij een hoger nummer.

Naast deze convertor hebben we ook een gekruiste seriële kabel of ook nullmodem kabel nodig. Met deze kabel verbinden we de twee compoorten met elkaar.

2. VIRTUAL SERIAL PORT

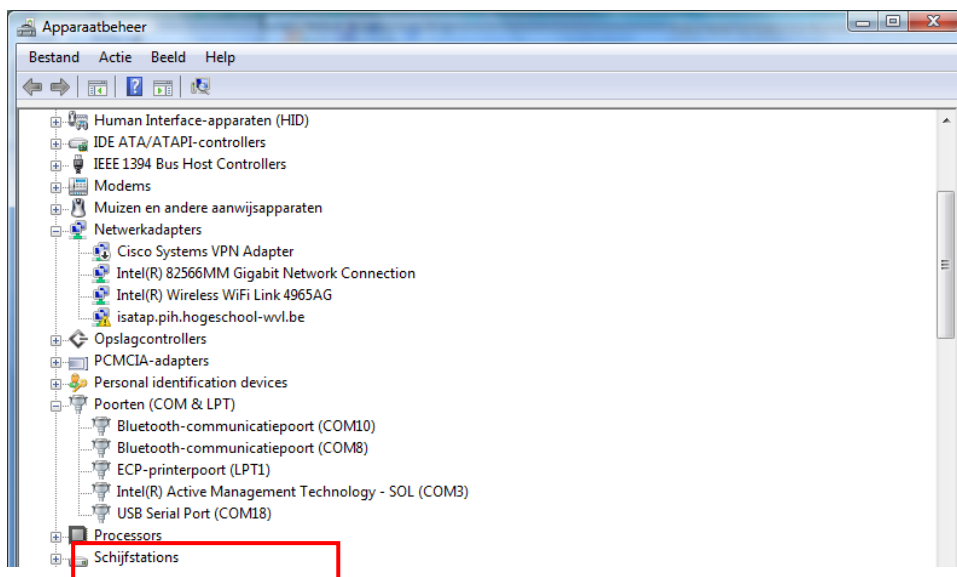
Deze software laat ons toe om een aantal seriële poorten te simuleren met een kleine driver. Met het programma kun je 2 virtuele seriële poorten maken die zogenaamd met elkaar verbonden zijn via een virtuele nullmodem kabel. Hierdoor kan je zender en ontvanger programmeren op dezelfde computer net alsof er een fysieke verbinding is tussen 2 computers met een nullmodem kabel. Je kan deze software downloaden op: <http://www.virtual-serial-port.com/downloads.html> of <http://www.eltima.com/products/vspdxp/>. Deze trial laat u toe om gedurende 15 dagen te werken met 2 virtuele seriële poorten die met elkaar verbonden zijn. Simuleer een null-modemkabel.



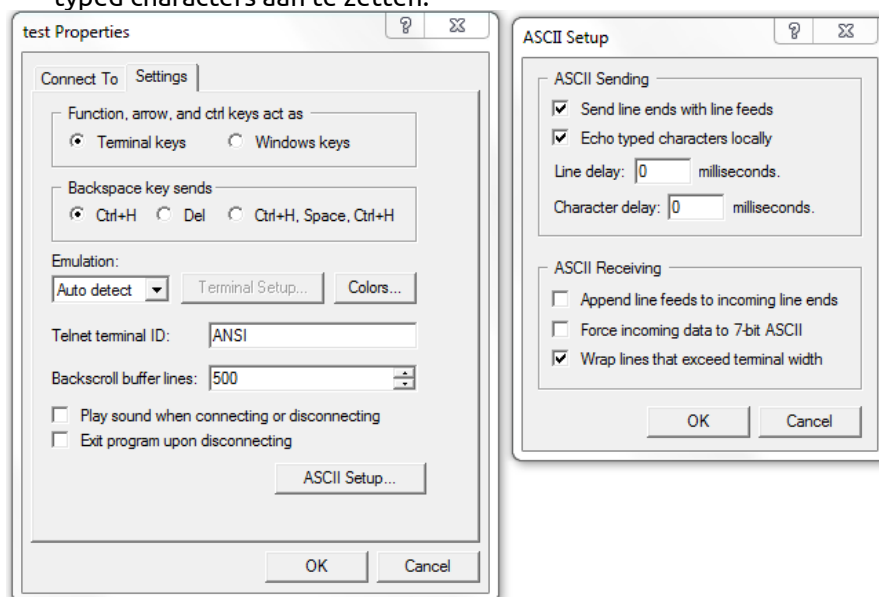
Opdracht seriële poort

Deze week kan je per twee werken via een nullmodem kabel en dien je het project als volgt in : NMCTx_loginnaam1_loginnaam2_serielepoort.rar, ofwel werk je alleen via de virtuele seriële poorten.

Installeer eerst hyperterminal. (map benodigdheden), installeer vervolgens de USB/RS232 convertor. Check in apparaatbeheer of de seriële goed is geïnstalleerd en noteer het com nummer van de poort.



- 1) Test de connectie, hiervoor open je elk een hyperterminal en controleer je of je data kan versturen en ontvangen. Zorg ervoor dat de instellingen van beide hyperterminals dezelfde zijn. Flowcontrol plaats je op geen d.w.z. geen handshaking. Handig is om de echo van de typed characters aan te zetten.



- 2) Eén hyperterminal wordt vervangen door een visual studio applicatie. Om een seriële poort te kunnen openen moet je een serialPort object . Via de properties kan je het serialPort object instellen. Zorg ervoor dat alle instellingen gelijk zijn met hyperterminal.
- 3) Toon alle beschikbare poorten in een combobox. Open vervolgens een gekozen poort. Sluit de poort terug bij het sluiten van de applicatie.
- 4) Verstuur tekst van de applicatie naar hyperterminal.
- 5) Verstuur tekst van hyperterminal naar de applicatie. Dit kunnen we doen op drie verschillende manieren. Maak een groupbox met drie radiobuttons, een radiobutton ReadExisting, een radiobutton Read en een radiobuttons dataReceived. We proberen om op de drie verschillende manieren de data te ontvangen. Voor de methodes Read en ReadExisting hebben we een timerke nodig. Om de 500ms voeren we de methode uit.
 - a. De methode ReadExisting() heeft de beschikbare bytes terug als een string.
 - b. De methode Read leest voortdurend het buffergeheugen van de seriële poort in. De methode Read() geeft ons een chararray terug die we zullen moeten omzetten naar een string. LET OP De string zit vol nullen als er niks is gelezen van de seriële poort. Als er enkele bytes zijn gelezen, dan worden die ook gevolgd door nullen. Om te weten hoeveel bytes er zijn binnengekomen, zul je dus een routine moeten schrijven die naar nullen zoekt. Als dit niet tijdig lukt komt er exception, die moeten we gewoon opvangen. Toch zijn er een aantal bytes gelezen.
 - c. Met het DataReceived() event hebben we geen timer meer nodig.
- 6) Nu bedenk je een protocol: Kies een karakter dat je zult gebruiken om duidelijk te maken dat je commando compleet is. Je zult in de aangekomen string dus moeten zoeken naar dat karakter. Als het commando compleet is, dan verwerk je het commando. Stuur een bevestiging naar de zender om duidelijk te maken dat de boodschap goed is aangekomen. Na de verwerking, maak je de commandostring terug leeg, en is die terug klaar om het volgende commando te gaan binnenlezen.
- 7) Bedien het USB daq board vanuit Hyperterminal. Maak instructies om de leds in groep en afzonderlijk te kunnen bedienen. Maak een instructie om de 2 analoge uitgangen te bedienen. Maak instructies om het LCD display te bedienen.
- 8) Als de status van een digitale of analoge ingang wijzigt moeten er een bericht naar hyperterminal verstuurd worden.