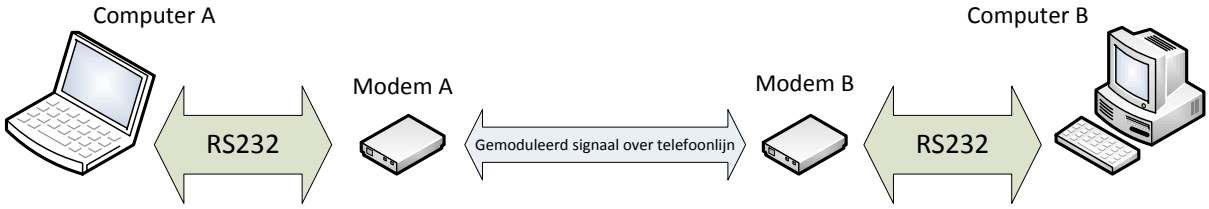
INHOUD

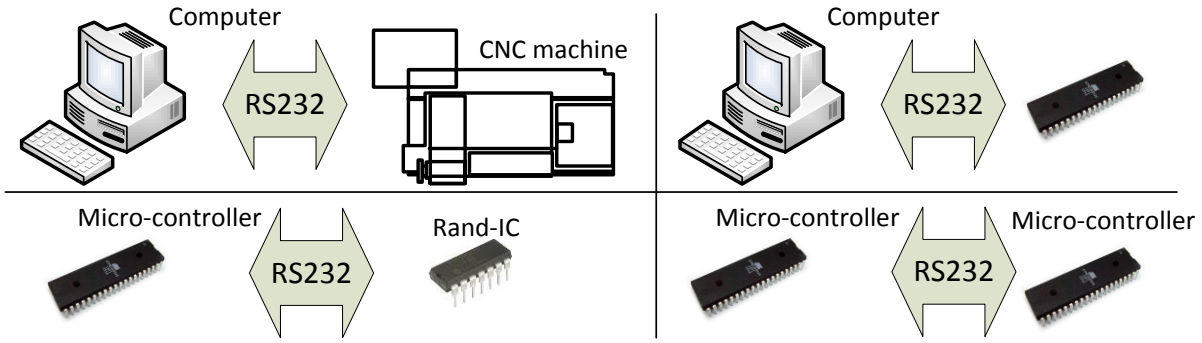
RS232

HISTORIEK

RS232 is een communicatie standaard die al sinds 1969 bestaat, maar vandaag nog steeds relevant is. Oorspronkelijk beschreef de standaard de communicatie tussen een computer en een modem, om zo over een telefoonlijn 2 computers met elkaar te kunnen verbinden. RS232 behandelde dus enkel de communicatie tussen de PC en de modem, maar met de opkomst van ADSL en breedband internet wordt deze vorm van communiceren met ouderwetse modems niet meer gebruikt.



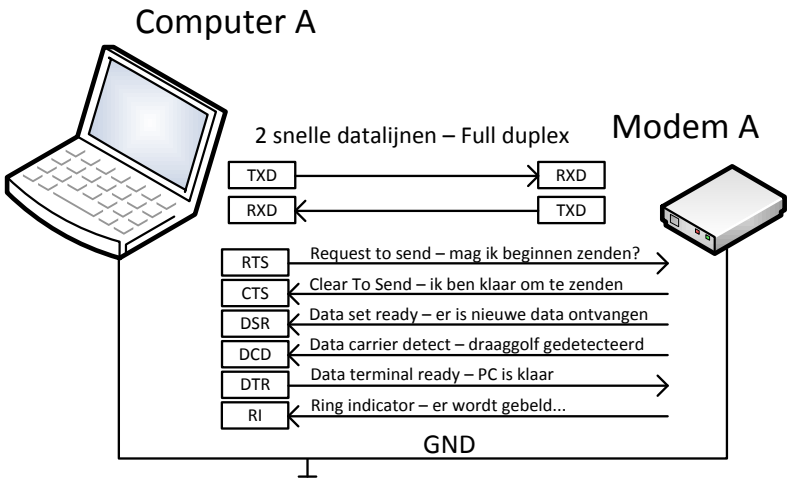
RS232 wordt nog wel steeds zeer veel gebruikt als communicatie standaard om data te communiceren tussen 2 digitale apparaten. Zo worden veel machines bestuurd door een PC – gebruik makend van het RS232 protocol en microcontrollers van amper 1€ kunnen zo communiceren via RS232 met computers, met rand-IC's of met andere microcontrollers.



De tegenwoordig officiële naam van de standaard is *ANSI/EIA/TIA-232-F, maar in het algemeen spraakgebruik leeft de term RS232 wel voort.*

RS232 – DE STANDAARD

AANSLUITINGEN



Zoals eerder aangehaald diende het oorspronkelijke RS232 protocol om computers te laten communiceren met modems, en daar vinden we nog steeds de sporen van terug.

De twee belangrijkste zijn de TXD en RXD lijnen. Over de TXD lijn wordt de data verzonden, over de RXD lijn wordt de data ontvangen. Merk op dat de TXD pin van de zender verbonden is met de RXD pin van de ontvanger. Bij moderne RS232 communicatie zijn deze RXD en TXD lijnen dikwijls de enige die nog gebruikt worden.

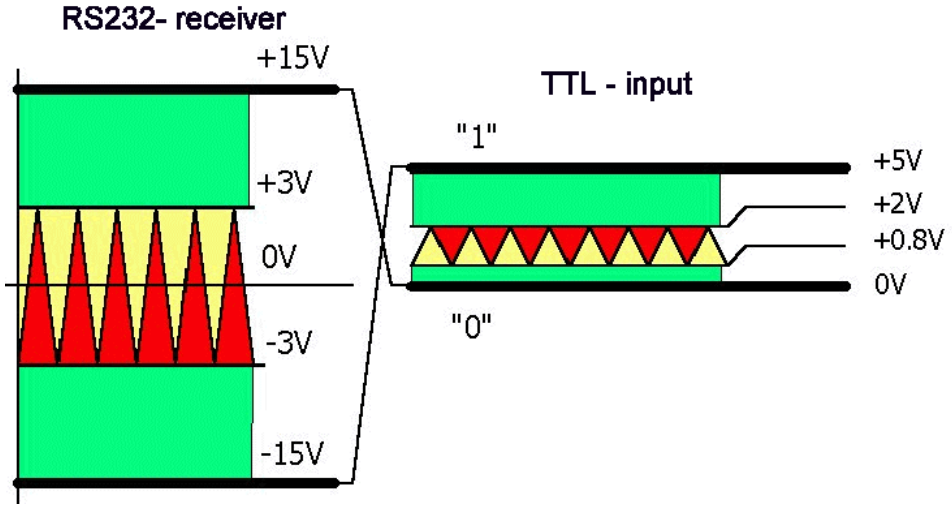
Naast de TXD en RXD lijnen beschreef de RS232 standaard ook een set van 'handshake lijnen'. Deze lijnen werden niet gebruikt om data op hoge snelheid te verzenden, maar dienden enkel als ondersteuning om een goede en betrouwbare communicatie op te zetten – in die tijd tussen PC en modem. Zo is er de RTS of request to send lijn die door de PC actief werd gemaakt om aan de modem te vragen of er nieuwe data mocht verstuurd worden. Als de buffer van de modem op dat moment leeg was en er was aan alle andere voorwaarden voldaan, dan maakt de modem de CTS (clear to send) lijn actief om aan de PC aan te geven dat de aanvraag werd goedgekeurd. De 6 meest gebruikte handshake lijnen zijn hier vermeld. Op de COM poort van een klassieke PC zijn deze handshake lijnen nog steeds beschikbaar.

CONNECTOREN

De originele standaard beschreef enkel de DB25 connector – die wij vandaag enkel nog kennen als de parallelle poort van een PC. De kleinere DB9 connector is vandaag het meest gebruikt als RS232 connector.

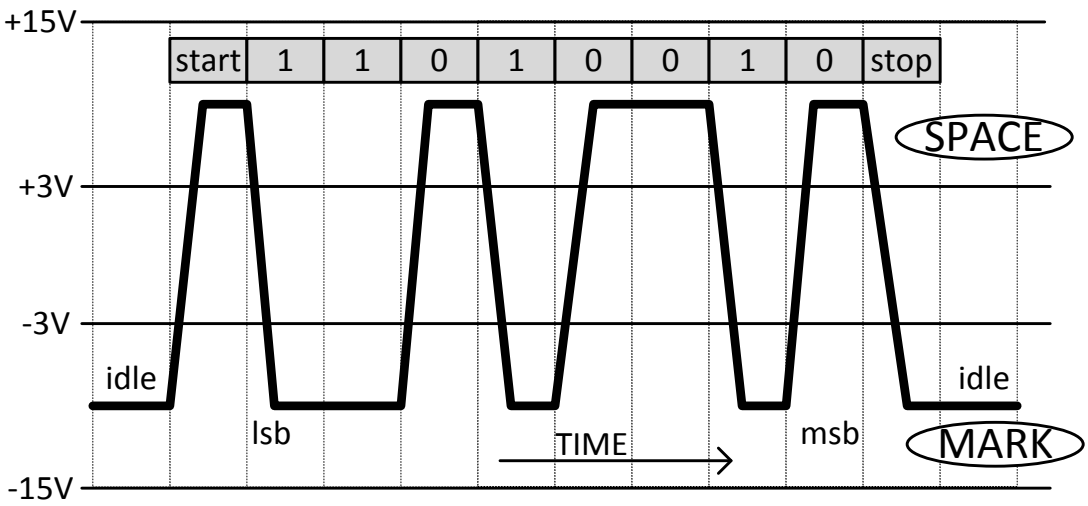
Moderne Pc's worden nagenoeg niet meer uitgerust met een com poort. Via een USB naar COM kabel kan je echter via een USB poort een COM poort emuleren.

SIGNAALNIVEAU ’S



De RS-232-standaard definieert de signaalniveaus die overeenkomen met een logische één of een logische nul als volgt:

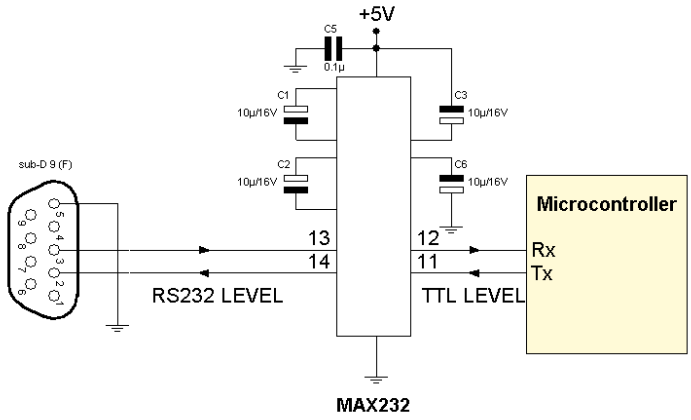
Logische één	AAN	MARK	tussen -3 en -15Volt
Logische nul	UIT	SPACE	tussen +3 en +15Volt



Deze symmetrische spanning – die dus nooit 0 volt kan worden – biedt een grotere storings-ongevoeligheid dan klassieke TTL signalen. Dit maakt dat RS232 over grotere afstanden data kan overdragen. Afhankelijk van de gebruikte voeding worden veelal signaalniveaus van ±5 V, ±10 V, ±12 V en ±15 V gebruikt. Ook de snelheid waarmee de signalen moeten veranderen (stijg- en daaltijd of *slew rate*) is vastgelegd in de norm.

MAX232

Een bekende omzetter om RS232 signaal – niveau's om te zetten van en naar TTL-niveau is de MAX232 van Maxim Semiconductors. Rond deze IC dienen enkele condensatoren aangesloten te worden. Deze worden gebruikt om de voedingsspanning van slechts 5Volt op te slingeren naar 12Volt.

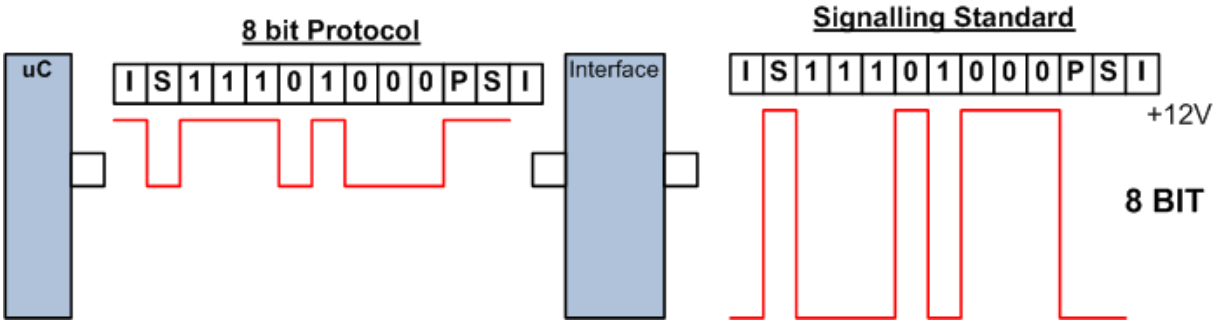


ASYNCHROON

U ziet dat er in de RS232 standaard geen plaats is voor een kloklijn. Dat maakt dat RS232 een asynchroon protocol is. Een asynchroon protocol houdt altijd in dat de synchronisatie tussen zender en ontvanger zal moeten geregeld worden via zeer duidelijke afspraken tussen die zender en ontvanger over o.a. bitsnelheid, startbit, stopbit, aantal bits, pariteit bit, …

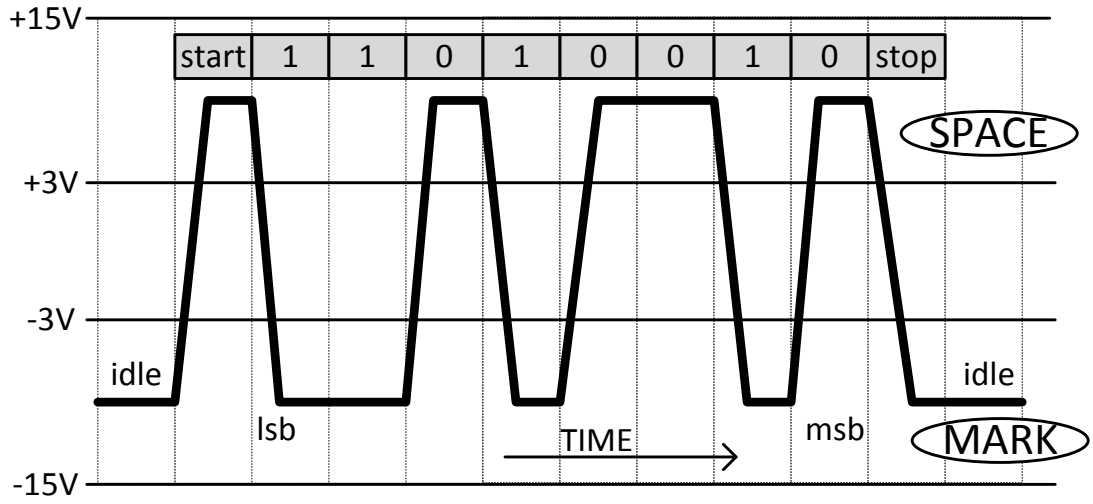
RS232 PROTOCOL

Een ander punt dat we hier zeker moeten bespreken is het protocol dat bij RS232 het meest wordt toegepast. Dit protocol maakt echter geen deel uit van de RS232 signaalstandaard, maar we bespreken het hier omdat dit protocol nagenoeg altijd wordt toegepast in combinatie met RS232.



Een **protocol** zegt iets over het aantal databits, start en stopbits, wel of geen pariteit, aantal bps, …

Een **signaalstandaard** beschrijft het aantal lijnen, de signaalniveau's, de gebruikte connectors, de slew-rate, …



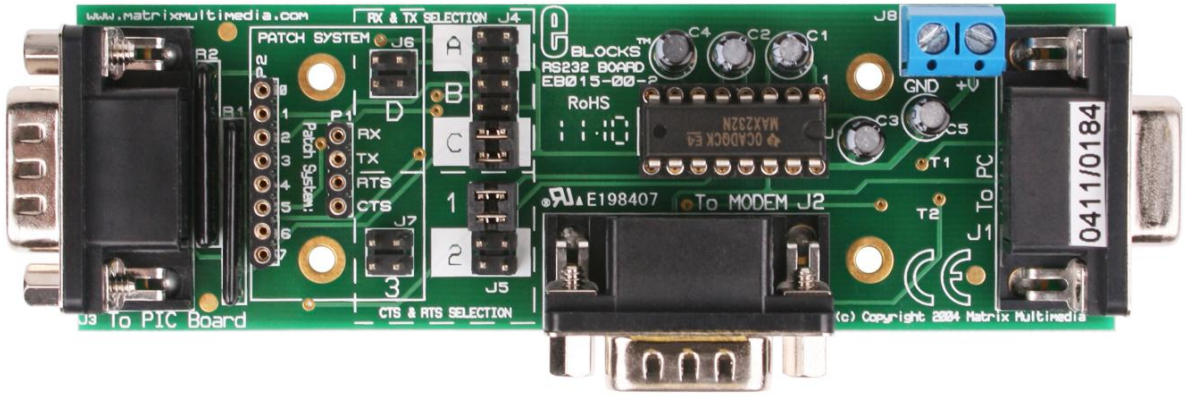
RS232 protocol…

- **STARTBIT**: Alle Asynchrone signalen hebben een start bit. Een start bit wordt gedetecteerd als het signaal van idle naar SPACE gaat en duurt één bitlengte.
- **AANTAL DATABITS:** In een aantal situaties kan je een keuze maken tussen 5,6,7,8 of zelfs 9 databits. Meestal kiezen we voor 8 databits.(5 keuzemogelijkheden)
- **PARITEIT:** Binnen het protocol kan er gekozen worden voor wel of geen pariteitsbit. Voorts kan er zelfs bepaald worden of het even of oneven pariteitscontrole is. (4 keuzemogelijkheden)
- **STOPBIT:** In een aantal gevallen kan er gekozen worden tussen 1, 1,5 of 2 stopbits. Meestal is dit gewoon 1 stopbit. (3 keuzemogelijkheden)
- **SNELHEID:** De datasnelheid wordt uitgedrukt in bps. Meestal heb je de keuze uit een aantal standaardwaarden: 110, 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600 (13 keuzemogelijkheden)

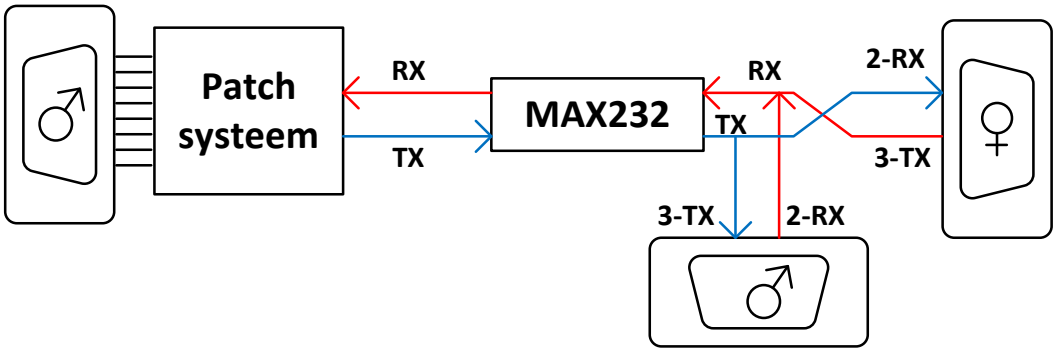
Zoals eerder vermeld is het bij asynchrone datatransmissie cruciaal dat de zender en ontvanger exact hetzelfde staan ingesteld. Stel voor dat de zender steeds 8 databits verzendt en dat de ontvanger er steeds slechts 7 verwacht…. Vermits dit protocol zoveel keuzemogelijkheden heeft is het zeer belangrijk dat je dit zeer nauwgezet instelt.

Er wordt over het RS232 protocol wel eens smalend gezegd dat 232 staat voor het aantal keren dat je moet proberen om de correcte combinatie te vinden. (5 x 4 x 3 x 13 = 780 verschillende combinaties mogelijk…)

E-BLOCKS RS232 MODULE (EB015):



Blokschema:

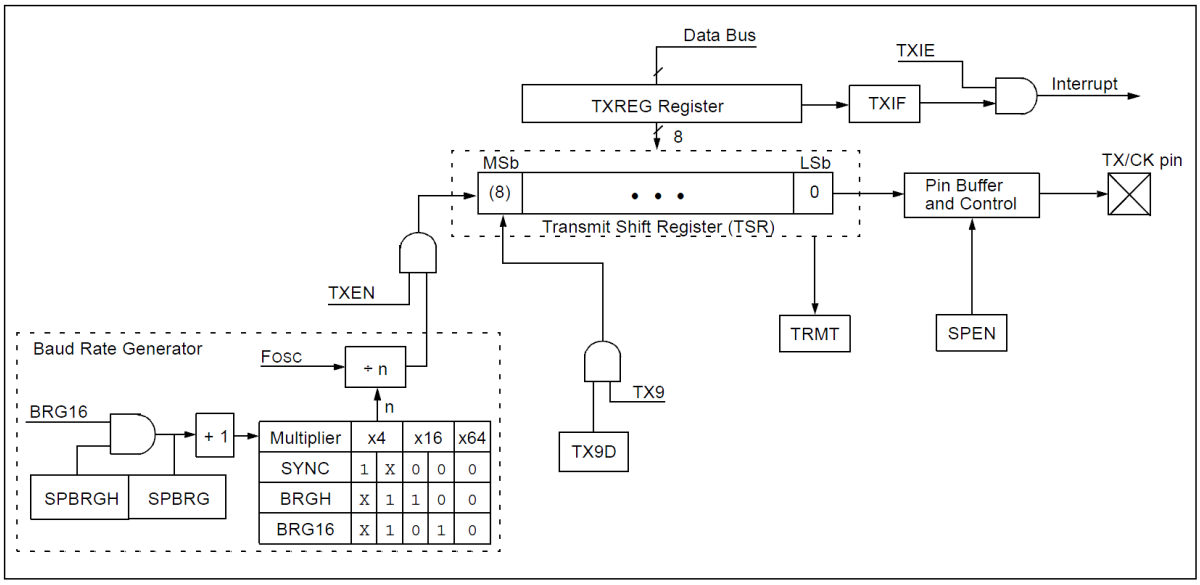


Dit bordje bevat als eerste een patch systeem om de RX en TX pinnen van de uC correct te verbinden met de RX en TX pinnen van de MAX232 IC. Deze MAX232 IC zal TTL signalen van de uC omzetten naar standaard RS232 signalen van + en – 12 Volt en omgekeerd van +-12V naar 5volt TTL signalen. Via 2 SUBD connectoren kan dit RS232 signaal dan worden doorgestuurd. Via J2 recht op recht naar buiten. Hier moet extern dan nog wel een crosslink kabel gebruikt worden die de TX met de RX lijn, en de RX met de TX lijn van de andere communicatie verbindt. Via de J1 connector is deze kruising reeds op het PCB gemaakt en kan deze module bijvoorbeeld recht op de COM poort van een PC worden aangesloten.

RS232 EN MICROCONTROLLER

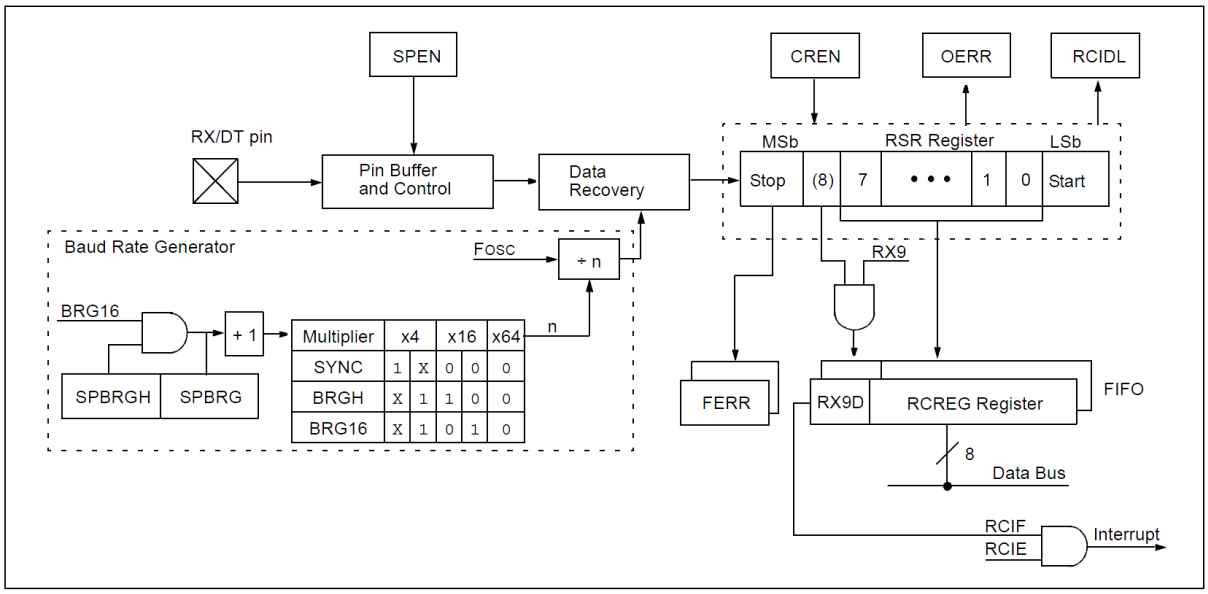
De meeste microcontroller hebben minstens één UART - of Universal Asynchronous Receiver Transmitter - aan boord die speciaal dient om een RS232 communicatie op te zetten. Het enige wat je moet doen is deze UART correct instellen via enkele registers. De te zenden data zet je in het TX register. De UART regelt vanaf dat moment dat deze data met de correcte snelheid, start en stopbit wordt verzonden. Ook voor het ontvangen van data neemt deze UART het meeste werk over zodat u zich in het programma met andere zaken kan bezig houden. Deze UARTS zijn geëvolueerd zodat ze nu ook synchroon data kunnen zenden en ontvangen. Vandaar de benaming EUSART "Enhanced Universal Synchronous and Asynchronous Receiver Transmitter"

FIGURE 12-1: EUSART TRANSMIT BLOCK DIAGRAM



U ziet hier het blokschema van de Transmitter van de UART van een PIC16F887. De baud rate generator moet correct worden ingesteld om – vanuit de processorklok – de correcte RS232 klok te genereren. Data die je wilt versturen zet je klaar in het 8 bit TXREG. De 9^{e} bit of eventueel de Pariteit bit die je zelf in de software reeds hebt bepaald – kan je in de TX9D bit klaar zetten. Van dit moment neemt de UART het van u over. Deze data zal met een correcte timing en toegevoegde start en stopbit serieel verstuurd worden via de TX pin. Indien gewenst kan een TX interrupt aangeven dat het TXREG terug leeg is zodat de volgende data klaar kan worden gezet. Deze instellingen zijn voor de meeste uC zeer gelijklopend. De respectievelijke datasheet begeleidt u stap voor stap door deze instellingen.

FIGURE 12-2: EUSART RECEIVE BLOCK DIAGRAM



Ook in het schema van de ontvanger moet de Baud rate generator correct worden ingesteld om de ontvangen signalen met de juiste snelheid te kunnen decoderen. De 8 – of eventueel 9 ontvangen data bits worden na dat de stopbit binnen is gekopieerd naar het RCREG en de 9^{e} bit wordt gekopieerd naar de RX9D bit. Als deze 9^{e} bit een pariteitsbit was, dan kan die in de software worden gecontroleerd. Een RC interrupt kan aangeven of er nieuwe data ontvangen is. Ook voor een correcte instelling van UART om RS232 signalen te ontvangen kan u best de datasheet van uw microcontroller raadplegen.

PROGRAMMA'S

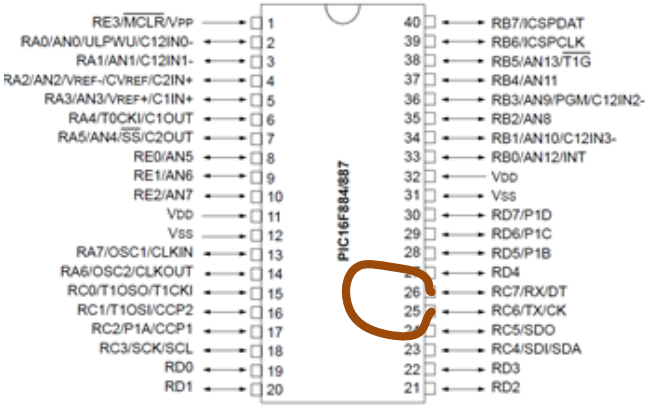
RS232 wordt gebruikt om data of getallen door te sturen, maar in de meeste gevallen wordt RS232 gebruikt om tekst of strings door te sturen van zender naar ontvanger. In de RS232 map enkele Flowcode en C programma – voorbeelden terugvinden.

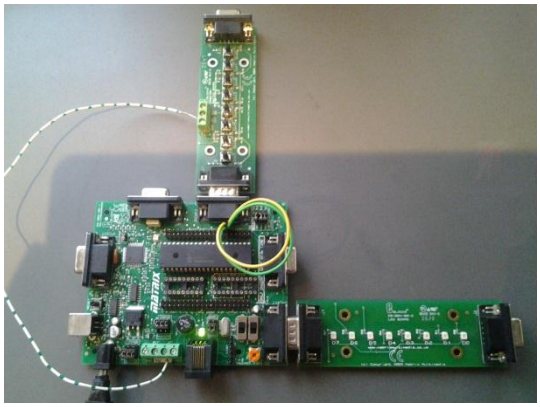
DEMO 1: ECHO - TX AAN RX - DATA IN C

HI-TECH C

data van switches via RS232 naar leds

Meetopstelling: Om deze programma's uit te testen is het niet altijd nodig om een afzonderlijke zender en ontvanger met elkaar te verbinden. Vermits UARTS full duplex werken, kunnen ze gelijktijdig zenden en ontvangen. Het volstaat dus om de TX pin te verbinden met de RX pin om zo de data die je uitstuurt meteen zelf terug te ontvangen.





PROGRAMMA ECHO_DATA IN C

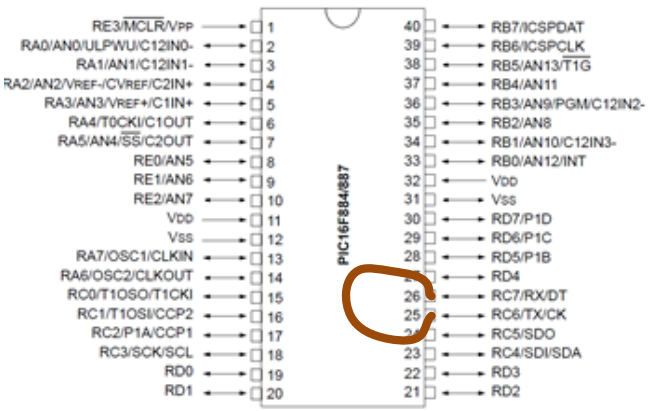
```
/*********************************************************

Bart Huyskens@telenet.be

RS232 DATA SEND and RECEIVE via RS232 UART

PIC16F887   External HS Xtal 19660800

PORTD = LEDS

PORTB = SWITCHES

!! connect TX (C6-PIN25) to RX (C7-PIN26)

*********************************************************/

#include <htc.h>

#define _XTAL_FREQ 19660800     // 19660800Hz - External HS Xtal

__CONFIG(FOSC_HS & WDTE_OFF & PWRTE_ON & MCLRE_ON & CP_OFF & CPD_OFF & BOREN_ON & IESO_ON & FCMEN_ON & LVP_OFF & DEBUG_OFF   );

__CONFIG(BOR4V_BOR21V & WRT_OFF);

void main() {

TRISB = 0xFF;           // All pins B = input for switches
```

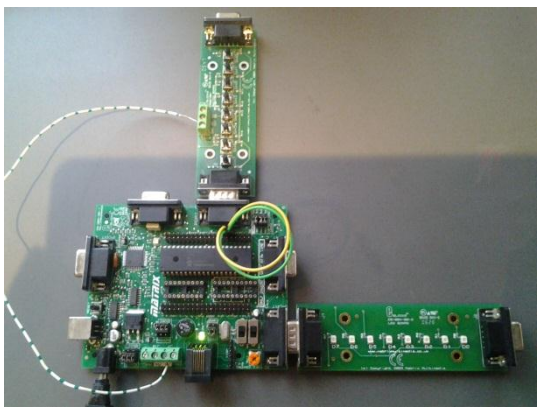
```
TRISD = 0x00;          // All pins D = output for leds
ANSEL = 0x00;          // Zet alle analoge ingangen uit
ANSELH = 0x00;         // Zet alle analoge ingangen uit
/**************SETTINGS UART**********************/
   // 511 is getal voor baud rate - Xtal 19.660.800 en 9600 bps      // FORMULE berekening baudrate: ((19660800 / 9600bps)/4)-1 = 511
   SPBRGH = 511 >> 8;              // vul baud rate register HIGH met 8 MSB's van spbrg
   SPBRG = 511 & 0xFF;             //  vul baud rate register LOW met 8 LSB's van spbrg
   TRISC6 = 0;                     // Pin C6 (pin 25) = TX als output zetten
   TRISC7 = 1;                     // PIN C7 (Pin 26) = RX als input zetten
   BAUDCTL = 0x08;                 // SCKP(0): Transmit non inverted data to TX pin / BRG16(1): 16 bit Baud rate gen
   RCSTA = 0x90;                   // SPEN(1): Serial Port Enabled / CREN(1): Enables Receiver
   TXSTA = 0x24;                   // TX9(0):8 bit / TXEN(1)Tansmit Enable / SYNC(0):Asynchr / BRGH(1): High Speed
/**************SETTINGS UART**********************/
while (1)   {
            while (!TXIF){}        //zolang TXIF (uit PIR1 register) bit laag is, mag er geen nieuwe data naar TXREG verstuurd worden.
            TXREG = PORTB;         // COPY switches to TXREG to send via TX pin
            while(!RCIF) {}        // wacht tot er nieuwe data beschikbaar is (1 = RCREG is full)(0 = RCREG = empty)
            PORTD = RCREG;         // COPY input data from RCREG to leds on PORTD
            }
}
```

DEMO 2: ECHO - TX AAN RX - DATA IN FC

FLOWCODE5 data van switches via RS232 naar leds (programma te vinden in RS232 map)

Meetopstelling: Om deze programma's uit te testen is het niet altijd nodig om een afzonderlijke zender en ontvanger met elkaar te verbinden. Vermits UARTS full duplex werken, kunnen ze gelijktijdig zenden en ontvangen. Het volstaat dus om de TX pin te verbinden met de RX pin om zo de data die je uitstuurt meteen zelf terug te ontvangen.

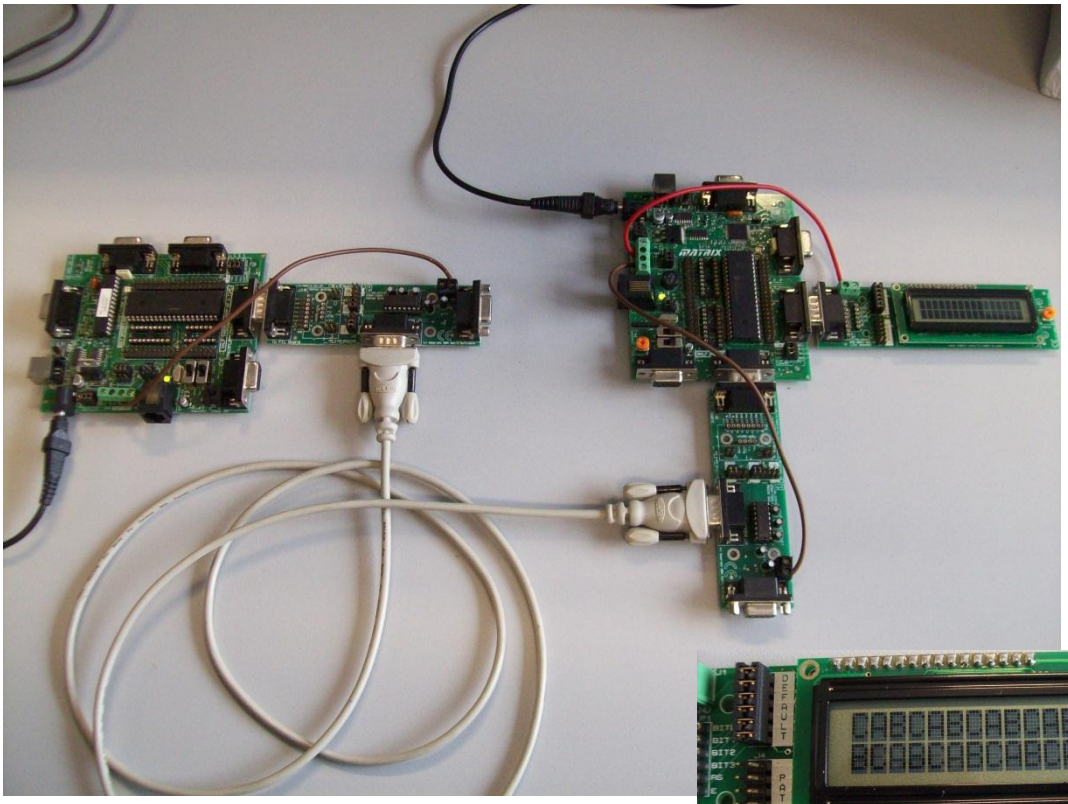




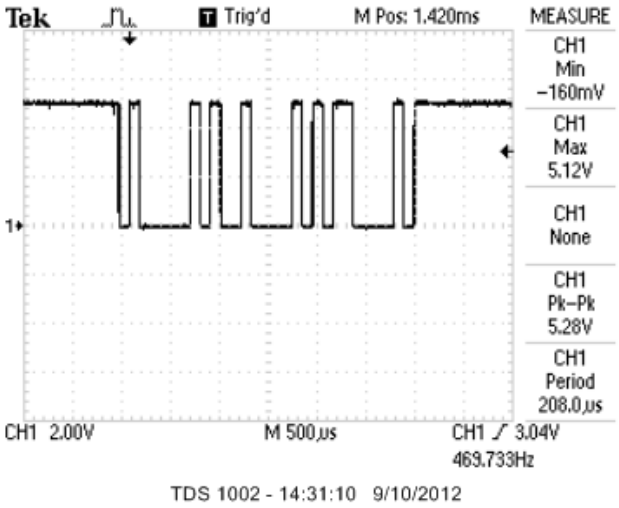
DEMO 3: ZENDEN EN ONTVANGEN VAN TEKST IN FC

FLOWCODE5 Zender stuurt in eeuwige loop de tekst “ABC” naar de ontvanger. De ontvanger toont de ontvangen karakters op het LCD. (programma te vinden in RS232 map)

Meetopstelling: Zender en ontvanger maken beide gebruik van het EB015 (met MAX232) bordje van Matrixmultimedia. Instellingen: POORTC – Jumpers op C – 2. (Soms kan de EB015 problemen veroorzaken tijdens het programmeren – even uittrekken is dan de oplossing). EB015 van zender en ontvanger verbinden via Cross-link kabel – TX aan RX en RX aan TX.



METINGEN AAN RS232 SIGNALEN

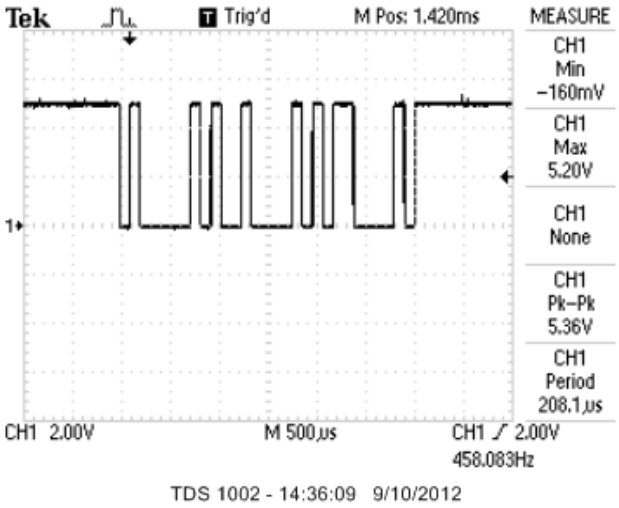


Meting aan “DEMO 3”.

Dit signaal is rechtstreeks gemeten op pin TX van de zendende microcontroller.

De tekst “ABC” wordt hier doorgestuurd.

Dit zijn nog steeds zuivere TTL signalen.

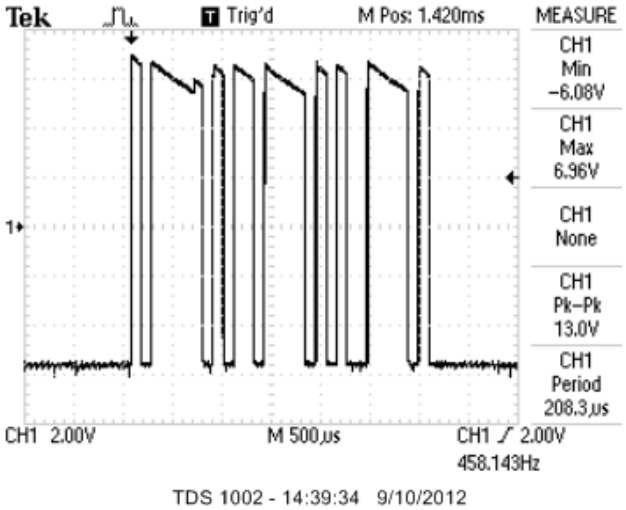


Meting aan “DEMO 3”.

Dit signaal is rechtstreeks gemeten op pin RX van de ontvangende microcontroller.

De tekst “ABC” wordt hier doorgestuurd.

Dit zijn nog steeds zuivere TTL signalen.



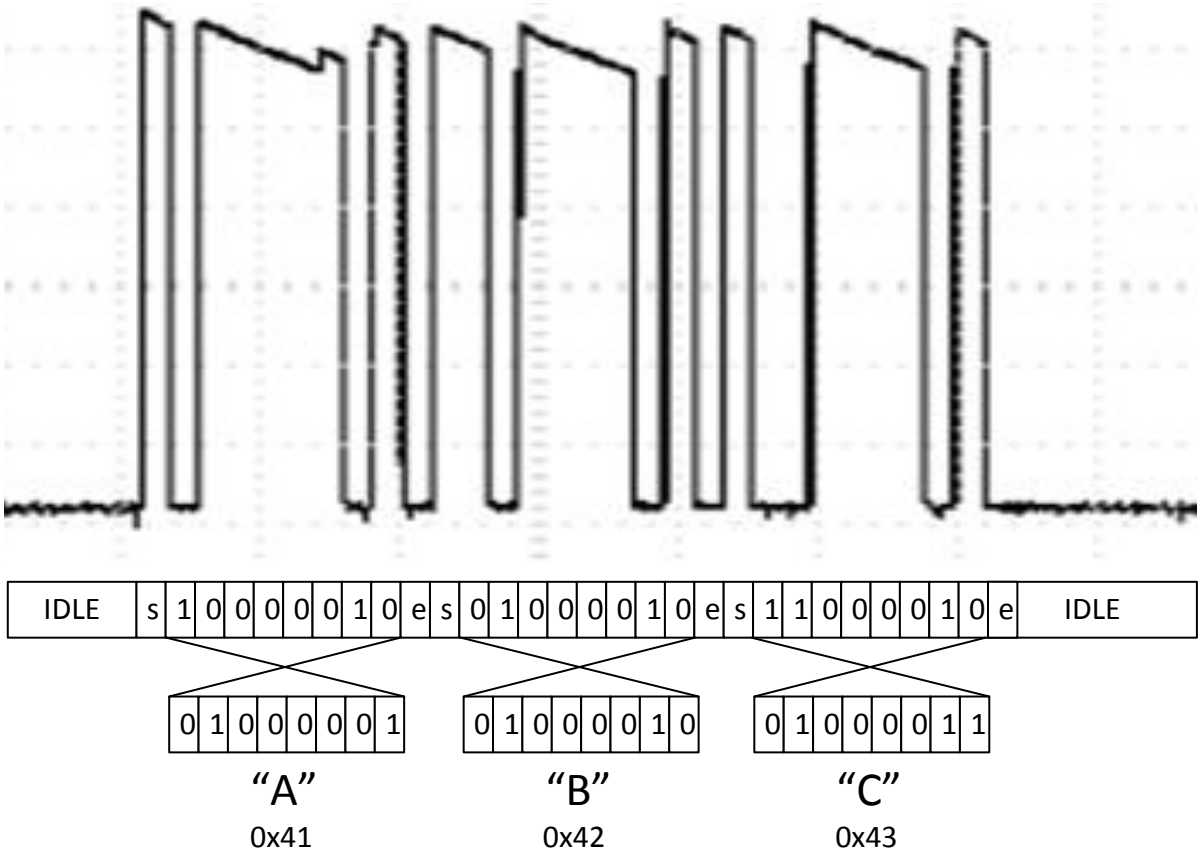
Meting aan “DEMO 3”.

Dit signaal is rechtstreeks gemeten op pin TX achter de MAX232 .

De tekst “ABC” wordt hier doorgestuurd.

U merkt op dat de signalen geïnverteerd zijn en wisselen tussen +7 en -6 volt.

ONTCIJFEREN VAN RS232 SIGNALEN



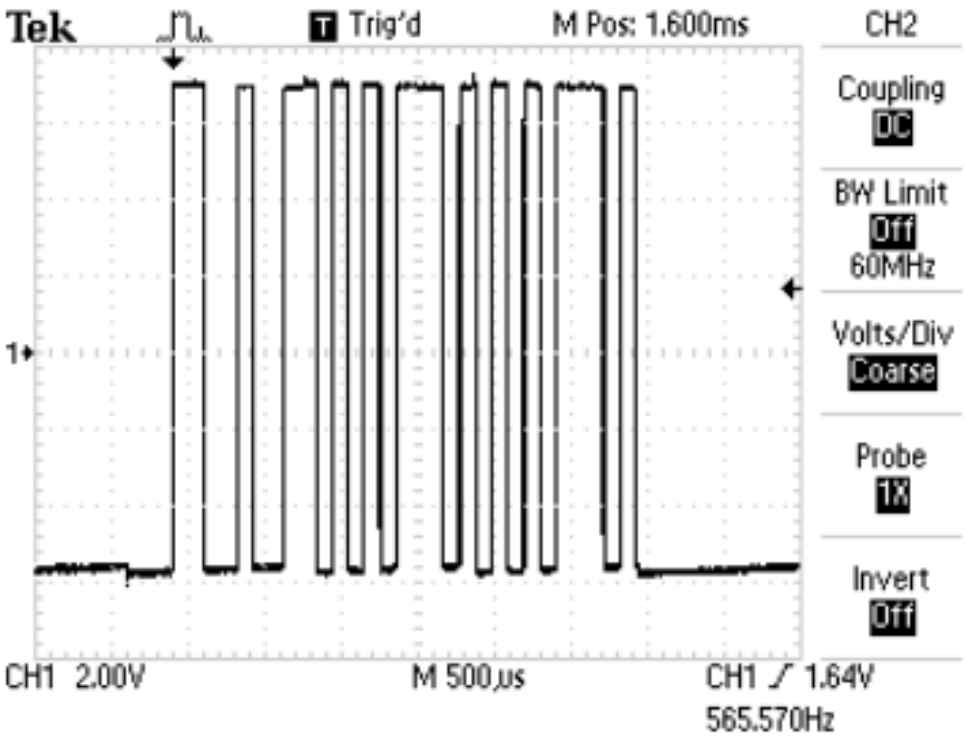
Bij RS232 signalen komt een +V overeen met een 0 en een –V stelt een 1 voor.

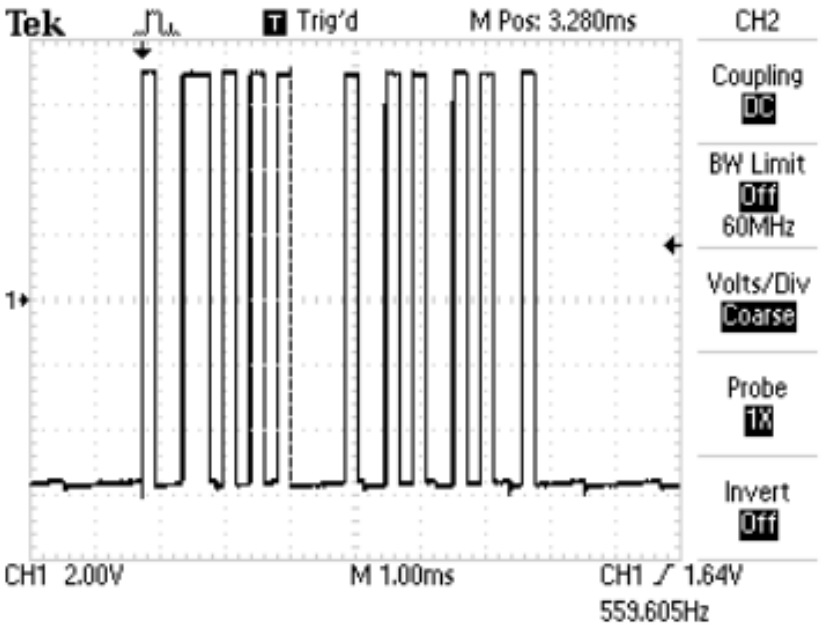
+V is in dit geval +7Volt en –V ongeveer -6Volt. Deze spanningen vallen mooi binnen de RS232 norm.

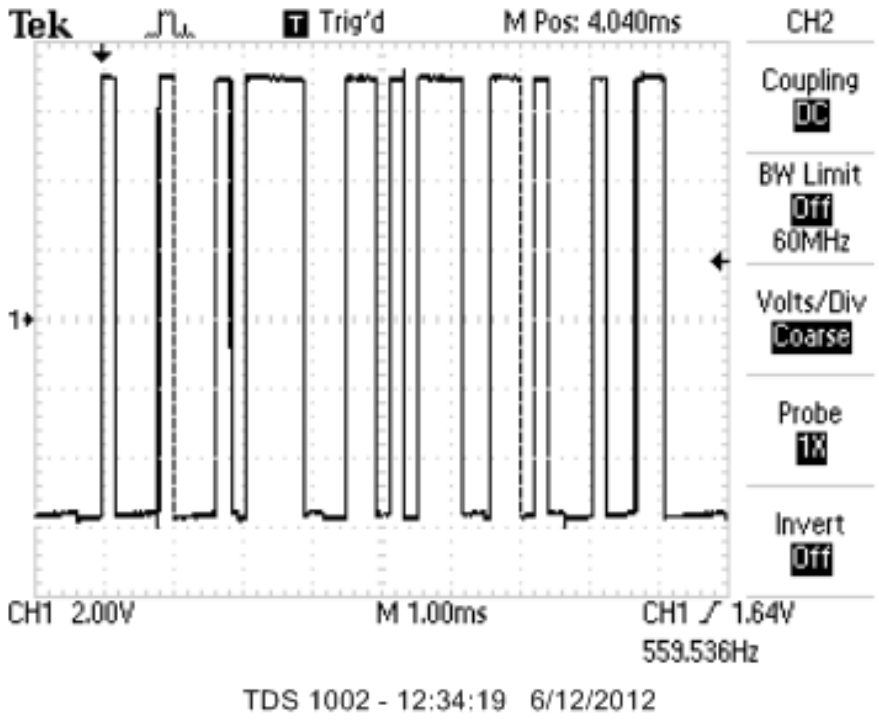
De zender zendt steeds de karakters “ABC” door. In de ascii tabel vinden we dat A overeenkomt met 0x41, B met 0x42 en C met 0x43. Elk karakter wordt voorafgegaan door een startbit “s” (0) en een stopbit “e” (1). Karakters worden steeds doorgezonden met hun LSB eerst, vandaar de omgekeerde karakters.

UITDAGINGEN

1. Leg uit wat het verschil is tussen een protocol en een signaalstandaard.
2. Zoek de datasheet op van een MAX232 – bekijk deze aandachtig. Teken het volledige schema op papier om deze te gebruiken tussen een microcontroller en de COM poort van een PC.
3. Wat is het nut van handshake lijnen.
4. Hoe lang duurt één frame met een RS232 communicatie met 1 start, 2 stopbits, 8 databits en één pariteitsbit tegen 9600 bps?
5. Hoe lang duurt het om tegen deze snelheid een foto van 1Mbyte door te sturen?
6. Vul de "datacom-fiche" in bijlage zo goed als mogelijk in.
7. Zet een RS232 communicatie op en test alle demoprogramma's
8. Zet een RS232 communicatie op met een klasgenoot en communiceer data met elkaar.
9. Zet een RS232 communicatie op met een klasgenoot en communiceer tekstberichten met elkaar.
10. Schrijf een programma dat uit het volgende tekstbericht: "X=516;Y=2256;Z=875" terug de 3 echte waarden uit deze tekst haalt. Gebruik van de ASCII tabel en kennis over string-manipulatie zal noodzakelijk zijn.
11. Ontcijfer onderstaande RS232 scoopbeelden. Welke karakters worden er hier doorgestuurd. Onderzoek ook op welke snelheid (in bps) de verschillende berichten werden doorgestuurd.




Tek
Trig'd
M Pos: 3.280ms
CH2
Coupling
DC
BW Limit
Off
60MHz
Volts/Div
Coarse
Probe
1X
Invert
Off
CH1 2.00V
M 1.00ms
CH1 ∕ 1.64V
559.605Hz
TDS 1002 - 12:30:34 6/12/2012


Tek
Trig'd
M Pos: 4.040ms
CH2
Coupling
DC
BW Limit
Off
60MHz
Volts/Div
Coarse
Probe
1X
Invert
Off
CH1 2.00V
M 1.00ms
CH1 ∕ 1.64V
559.536Hz
TDS 1002 - 12:34:19 6/12/2012

DATACOM-FICHE

Naam:…… Klas: …….. Nr:…….. Datum:……..

Te bespreken protocol: ……

Algemeen:

Parallel/Serieel		Karakterbeveiliging	pariteitsbit ? …?
Synchroon/Asynchroon		Karaktersynchronisatie	Startbit? Stopbit? …?
Simplex/half/Full duplex		Berichtbeveiliging	CRC? LRC/VRC? …?
Bitstuffing	Zoja - hoe?	Berichtsysnchronisatie	TSX ? , ETX ?, ….?
Hardware in uC en/ of bitbanging	Zit er hardware in de uC die deze communicatie rechtstreeks kan genereren? Zoja - plaats hier dan een blokschema. Zou bitbanging ook mogelijk zijn?		

7 lagen model

Application	Is er een applicatie die de link vormt tussen mens en machine - heb je deze zelf geschreven? In welke taal? Naam programma. Welke link is er met de mens?
Presentation	Wordt de informatie uit de applicatie op één of andere manier vertaald naar een standaardformaat? Is er een besturingssysteem aanwezig?
Session	Is er een module die op bepaalde momenten de communicatie opzet en op andere momenten weer terug afbreekt?
Transport	Is er een bepaald systeem opgezet zodat de zender bevestiging krijgt dat de ontvanger de data goed ontvangen heeft?
Network	Is er een module die bepaalt welke route de data moet volgen om van zender naar ontvanger te geraken.
Data link	Wordt de data opgedeeld in pakketten of frames? Hoe groot zijn deze frames? Startbit, stopbit, pariteitsbit, CRC controle,
Physical	Welke kabels worden er gebruikt, hoe ver kan er maximaal gezonden worden? Welke spanningen worden er gebruikt? Datasnelheid? Wordt er gemoduleerd? Welke connectoren? Return to zero/non return to zero/differentieel/…, …

Metingen op signaal:

Meet met de oscilloscoop en/ of de logic analyser de datacommunicatie op en zet de beelden hier. Ontcijfer eventueel de data. Voorzie de beelden van een beetje uitleg.

(verwijder al de oranje tekst - deze dient enkel als hulp)