

NeoPixel (UART + DMA)

Donnerstag, 27. November 2025

14:31

Serielle Bussysteme

Allgemeine Fakten

Die Kommunikation zwischen zwei Systemen bedingt, dass diese ein identisches Verständnis von der Zeit haben, so dass der Sender mit einem vorgegebenen Intervall Daten ausgeben/senden und der Empfänger dieses mit dem identischen Intervall abtasten/empfangen kann.

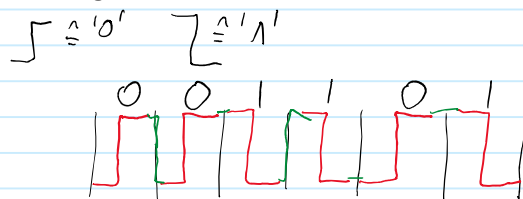
Zur Darstellung gibt es zwei möglich Verfahren:

Synchrone Datenübertragung:

Parallel zum Datensignal gibt es ein Taktsignal, welches dem Sender und Empfänger mitteilt, wann Daten auf den Datenbus zu schreiben/zu lesen sind. Dieses Taktsignal kann gleichermaßen als eigenständige Leitung aber auch im Datensignal kodiert sein (Leitungscode).

Beispiel Manchesterkodierung (Ethernet 100MBit)

Bei der Manchesterkodierung entspricht eine Null-Eins Folge einer logisch 0 und eine Eins-Null Folge einer logischen 1



Asynchrone Datenübertragung:

Hier gibt es kein separates Taktsignal, stattdessen senden/empfangen die Teilnehmer mit ihrer eigenen 'Uhr' und nutzen den Fakt aus, dass für kleine Zeitabschnitte der Gangunterschied tolerierbar ist.

Zur Synchronisation der Uhren wird ein Startsignal ausgesendet (bei UART die fallende Flanke des Startbits), zu welchen beide Teilnehmer ihre Uhr auf 0 setzen. Die max. Paketlänge ergibt sich aus der tolerierbaren Taktabweichung.

Sender und Empfänger müssen 'händisch' auf die identische Taktrate eingestellt werden. Da die Prozessorfrequenz und die Baudrate nicht ganzzahlige Vielfache sind, sind allein deswegen Taktabweichungen vorhanden, so dass die Datenlänge zumeist sehr kurz ist.

UART Universal Asynchron Receiver/Transmitter

In Verbindung mit UART sind div. weitere Begriffe verbunden

RS232

Chip für USB zu UART Konvertierung --> Das wäre der FT232 Baustein von FTDI, der hierzu gerne genutzt wird und im Namen den Begriff 232 trägt. Dieser beinhaltet einen USB nach UART Konverter, der Ausgabepegel ist jedoch TTL

RS Recommend Standard 232

Definiert Stecker DB9/DB25

Basiert auf UART Datenübertragung

Definiert die Datenübertragung über Spannungen

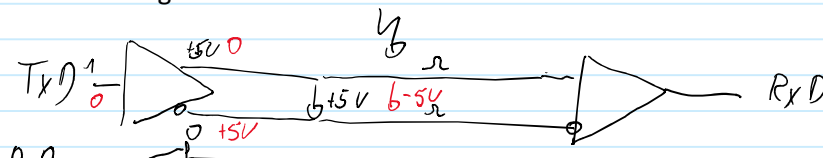
'0' -> +3V...+15V

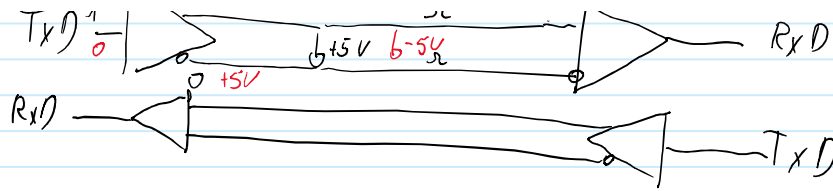
'1' -> -3V...-15V

RS422

Basiert auf UART Datenübertragung

Die Sende und Empfangsleitung ist eine zweiadrigte Datenleitung, auf welcher die Daten Differentiell gesendet werden

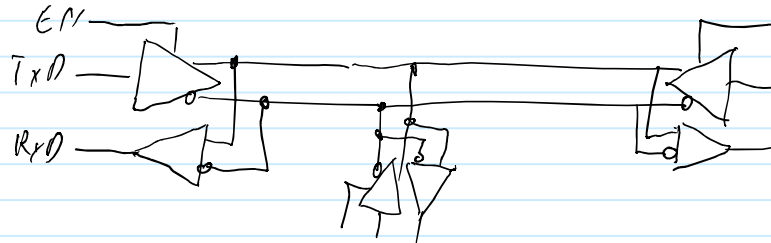




Ähnlich zu RS232, nur dass anstatt der unipolaren Spannung nun eine Differentielle Spannung übertragen wird.

Identisch zu RS232 ist eine Adressierung der Teilnehmer nicht notwendig (Punkt zu Punkt Übertragung)

RS485



Ähnlich zu RS422, nur dass der Transceiver deaktiviert werden kann und somit mehrere Teilnehmer angeschlossen werden können.

HalbDuplex Übertragung!

--> Im NXT Baustein vorhanden an SensorPort 4, bei welcher der Treiber mit PA7 aktiviert/deaktiviert wird

UART

gemeinsame Baudrate (Taktrate)

1200, 2400, 4800, 9600 Baud, 115200Baud

Rahmenformat

- Startbit (fest 1 Bit, welches 0 ist)
- Datenbits (5,6,7,8,9) Datenbits
- Parity (Optional, Prüfsumme, so dass der Empfänger prüfen könnte, ob die Datenbits korrekt übertragen wurden)

☐ Not Parity

☐ Even Parity: Drive to 0 when the number of 1 in data ist even

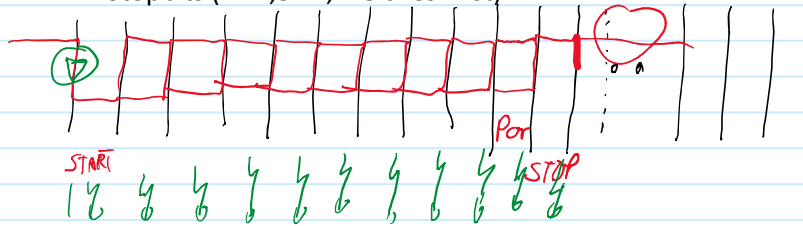
0 1 0 1 1 0 0 0 1
0 1 0 1 1 0 0 1 0

Odd Parity: Gegenteil zu Even

Mark Parity: Paritätsbit ist immer 1

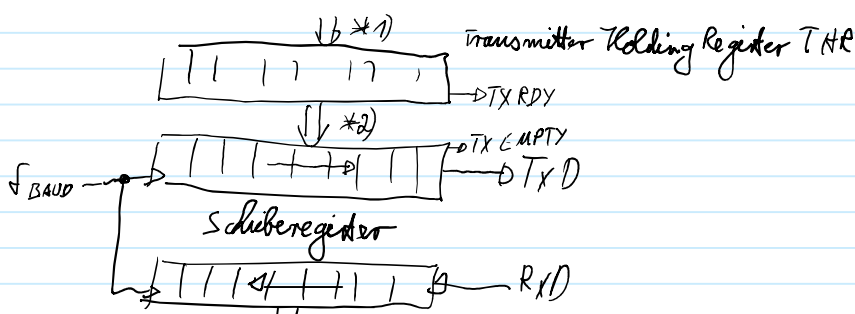
Space Parity: Paritätsbit ist immer 0

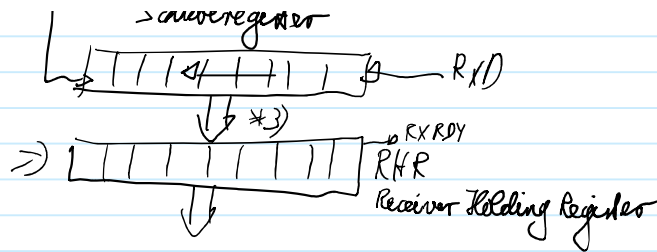
- Stopbits (1 1,5 2, welches 1 ist)



UART gibt nur dieses Rahmenformat vor. Welche Bedeutung die übertragenen Bytes hat, hängt vom nächst höheren Protokoll ab.

Der prinzipielle Aufbau einer seriellen Schnittstelle sieht wie folgt aus





- *1) In das TransmitterHoldingRegister THR darf nur geschrieben werden, wenn dieses Leer ist (erkennbar an TXRDY), andernfalls würde der alte Wert überschrieben und nicht gesendet werden. Wenn beim Schreiben in das THR Register das Schieberegister ebenfalls leer ist, wird der Inhalt sofort in das Schieberegister übertragen und das THR Register könnte den nachfolgend zu sendenden Wert aufnehmen.
- *2) Nach senden des Stopbits prüft das Schieberegister, ob ein neuer Wert im THR steht. Wenn ja, wird dessen Inhalt in das Schieberegister zum Senden übertragen. Wenn nein, wird der Sendeprozess beendet und ein Idle Signal (Pegel = 1) gesendet.
- *3) Wenn das Stopbit empfangen wurde und die Prüfsumme (sofern vorhanden) korrekt ist, wird der Inhalt aus dem Schieberegister in das ReceiverHoldingRegister RHR übertragen. Wenn das RHR nicht rechtzeitig gelesen wird, würde mit diesem Schreiben der zuvor empfangene Wert überschrieben/verloren gehen.

Figure 31-9. Transmitter Status

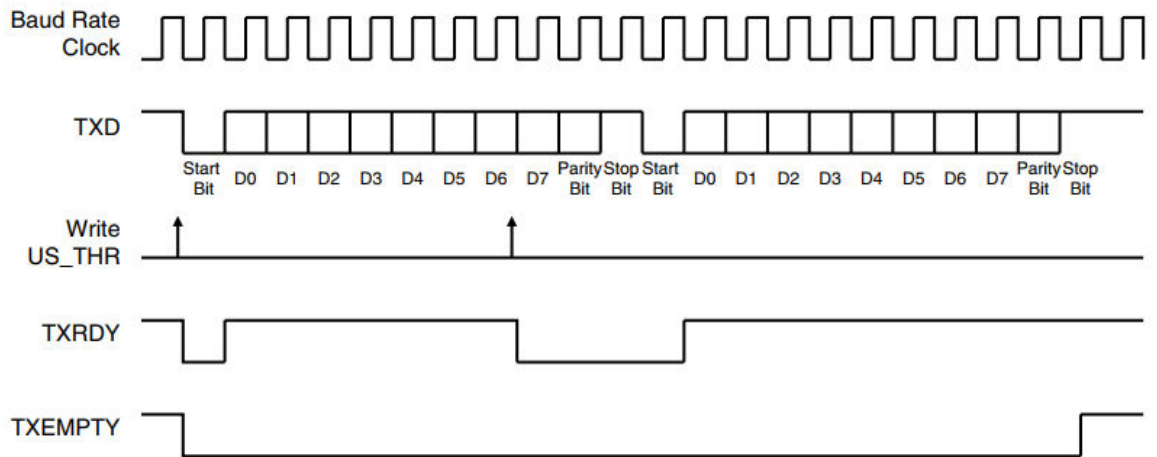


Figure 31-13. Receiver Status

