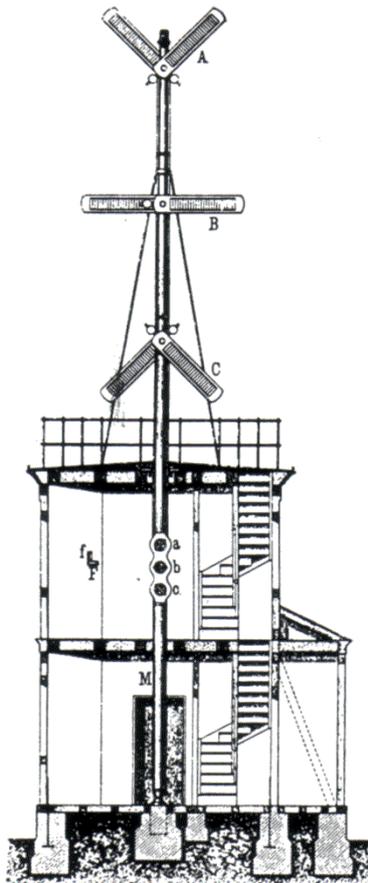


Übungsblatt 8

Abgabe: 15.12.2011

Aufgabe 1 – Datentransferrate und Latenz



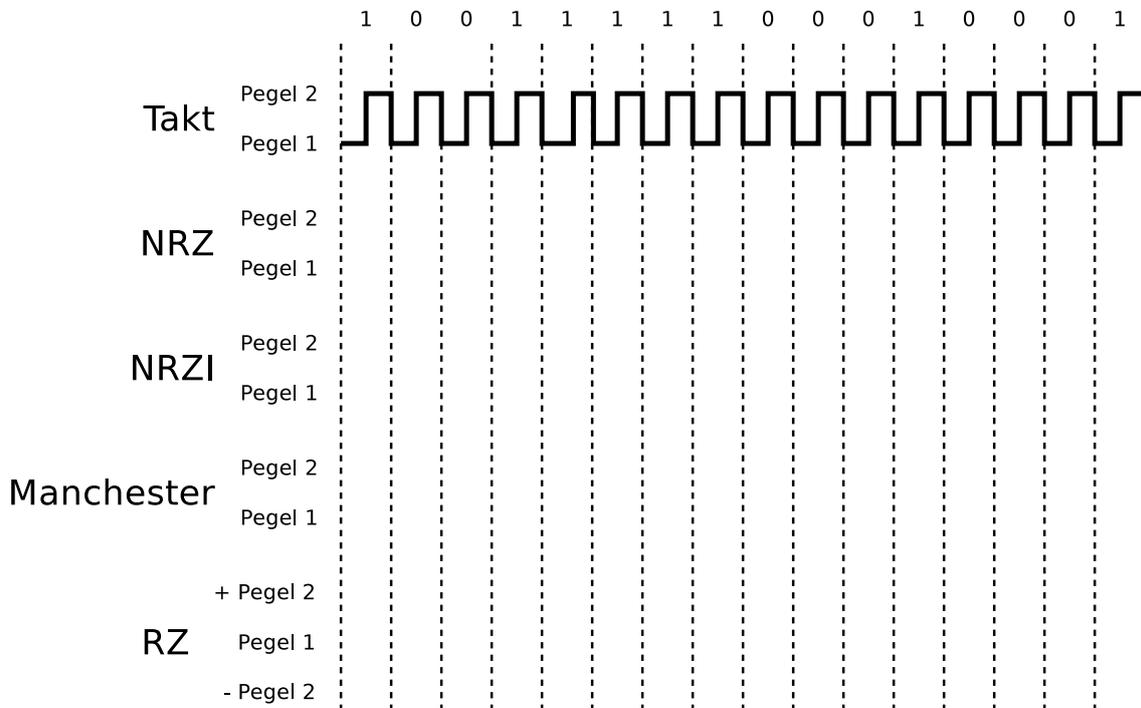
Der Preußische optische Telegraf (1832-1849) war ein telegrafisches Kommunikationssystem zwischen Berlin und Koblenz in der Rheinprovinz. Behördliche und militärische Nachrichten konnten mittels optischer Signale über eine Distanz von fast 550 km via 62 Telegrafstationen übermitteln werden. Jede Station verfügte über 6 Telegrafenarme mit je 4 Positionen zur Kodierung.

1. **Datentransferrate:** Wenn man alle 10 Sekunden eine neue Einstellung der Telegrafenarme vornehmen kann, wie viele Bit können pro Sekunde übertragen werden?
2. **Latenz:** Wenn jede Station 1 Minute für die Weiterleitung benötigt, wie groß ist die Ende-zu-Ende-Verzögerung?

Aufgabe 2 – Kodierung (Bitübertragungsschicht)

1. Geben Sie die gesuchten Kodierungen für das Bitmuster an.

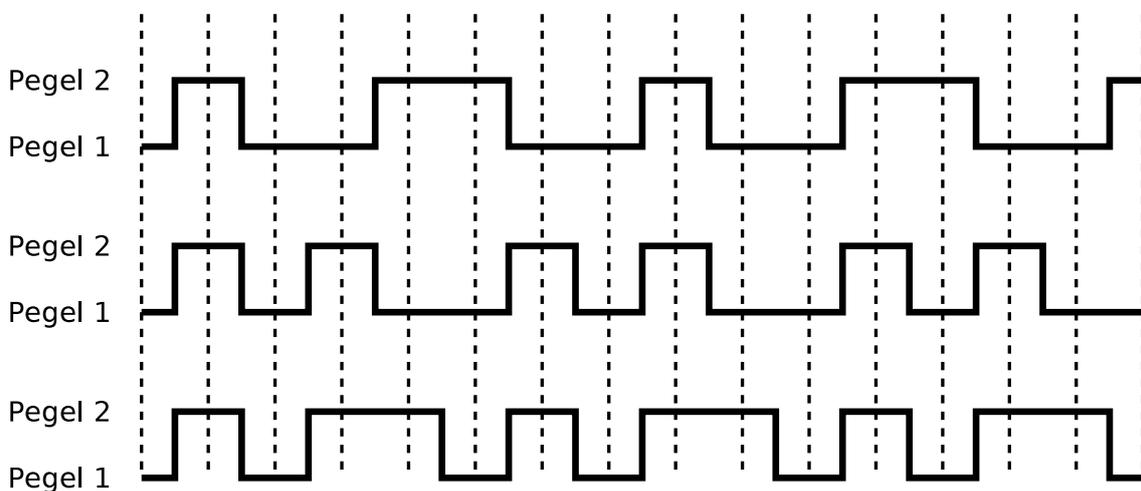
- Gehen Sie davon aus, dass das NRZI-Signal auf low beginnt.



2. Kodieren Sie die folgenden beiden Bitfolgen mit 4B5B und NRZI und geben Sie den Signalverlauf an.

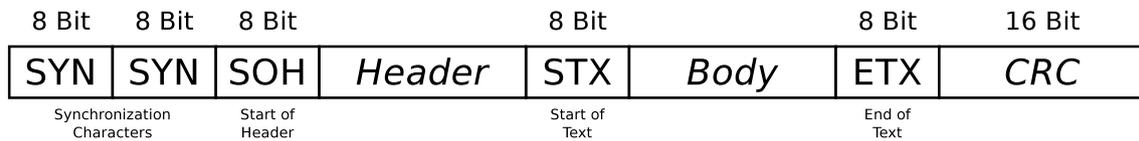
- 0010 1111 0001
- 1101 0000 1001

3. Folgende Signalverläufe sind mit NRZI und 4B5B kodiert. Geben sie die Nutzdaten an.



Aufgabe 3 – Zeichenstopfen (Sicherungsschicht)

Die Sicherungsschicht übernimmt den Bitstrom von der Bitübertragungsschicht und unterteilt diesen in Rahmen (sog. Frames). Ein älteres, synchrones, zeichenorientiertes Verbindungsprotokoll von IBM ist BISYNC (Binary Synchronous Communication). BISYNC verwendet eine definierte Folge von Steuerzeichen für die Synchron-Übertragung von binär codierten Daten. BISYNC-Rahmen sind wie folgt aufgebaut:



Steuerzeichen markieren die Struktur des Rahmens. Kommen bestimmte Steuerzeichen (ETX und DLE) im Nutzdatenteil (Body) vor, werden sie von der Sicherungsschicht durch ein DLE-Zeichen geschützt (\implies Zeichenstopfen).

Steuerzeichen	SOH	STX	ETX	DLE	SYN
HEX-Wert	01	02	03	10	16

Ein ETX im Nutzdatenteil wird durch DLE ETX kodiert. DLE selbst muss durch DLE DLE kodiert werden.

1. Wie lauten die Nutzdaten der folgenden BISYNC-Rahmen?

- 16 16 01 99 98 97 96 95 02 A1 A2 A3 A4 A5 03 A0 B7
- 16 16 01 99 98 97 96 95 02 01 02 10 03 04 05 03 76 35
- 16 16 01 99 98 97 96 95 02 10 03 10 10 10 03 03 92 55

Aufgabe 4 – Fehlerkorrektur (Sicherungsschicht)

Die Existenz von Übertragungsfehlern kann mit CRC-Prüfsummen nachgewiesen werden. Sollen Fehler nicht nur erkannt, sondern auch korrigiert werden können, müssen die zu übertragenden Daten entsprechend kodiert werden. Fehlerkorrektur kann man mit dem **Hamming-Abstand** realisieren. Vergleicht man 2 Bitfolgen, kann man feststellen, wie viele Bits nicht übereinstimmen.

Beispiel: 10100110 und 10110101. In diesem Fall sind 3 Bit verschieden.

Um festzustellen, wie viele Bits nicht übereinstimmen, verknüpft man beide Bitfolgen mit XOR und zählt die Einsen im Ergebnis. Der Hamming-Abstand ist die Anzahl der Bitpositionen, in denen sich 2 Bitfolgen unterscheiden.

• Schritt 1: Position der Prüfbits

- Die Bits einer Bitfolge werden beginnend mit 1 von links durchnummeriert. Bits, die Potenzen von 2 sind (1, 2, 4, 8, 16, usw.) sind Prüfbits. Die übrigen Bits sind die Nutzdaten.
- Beispiel: 8 Bit Nutzdaten: 01001100

```

      Position:  1  2  3  4  5  6  7  8
      Nutzdaten:  0  1  0  0  1  1  0  0
  
```

```

      Position:  1  2  3  4  5  6  7  8  9 10 11 12
zu übertragende Daten:  ?  ?  0  ?  1  0  0  ?  1  1  0  0
  
```

- Nun müssen noch die Werte der Prüfbits ermittelt werden.

• Schritt 2: Werte der Prüfbits ermitteln

- Die Werte der Prüfbits muss der Sender ermitteln. Jeder Bit-Position in der Übertragung wird die Positionsnummer zugeordnet. Die Positionsnummer ist in diesem Beispiel 4-stellig, da wir 4 Prüfbits haben.

Beispiele:

Position: 1 \implies Wert: 0001

Position: 2 \implies Wert: 0010

Position: 3 \implies Wert: 0011

Position: 4 \implies Wert: 0100

Position: 5 \implies Wert: 0101

...

- Als nächstes werden die Werte derjenigen Nutzdatenpositionen, die 1 in unserer Übertragung sind, mit XOR zusammen gerechnet. Im Beispiel Positionsnummer 5, Positionsnummer 9 und Positionsnummer 10.

```

      Position:  1  2  3  4  5  6  7  8  9 10 11 12
zu übertragende Daten:  ?  ?  0  ?  1  0  0  ?  1  1  0  0
  
```

```

      0101  Position 5
      1001  Position 9
XOR 1010  Position 10
-----
      = 0110
  
```

- Das Ergebnis sind die Werte der Prüfbits. Diese werden in die Übertragung eingefügt.

```

      Position:  1  2  3  4  5  6  7  8  9 10 11 12
zu übertragende Daten:  0  1  0  1  1  0  0  0  1  1  0  0
  
```

• Schritt 3: Empfangene Daten prüfen

- Der Empfänger kann in der Sicherungsschicht überprüfen, ob eine empfangene Bitfolge korrekt ist. Dafür berechnet der Empfänger selbst auch die Prüfbits zu den empfangenen Nutzdaten und vergleicht diese mit den empfangenen Prüfbits. Prüfbits befinden sich auf den Positionen 1, 2, 4 und 8.

```
empfangene Daten: 1  2  3  4  5  6  7  8  9 10 11 12
                  0  1  0  1  1  0  0  0  1  1  0  0
```

```

      0101   Position 5
      1001   Position 9
XOR 1010   Position 10
-----
      0110   Prüfbits berechnet
XOR 0110   Prüfbits empfangen
-----
=   0000   => Korrekte Übertragung
```

- Wenn etwas bei der Übertragung schief gelaufen ist...

```
empfangene Daten: 1  2  3  4  5  6  7  8  9 10 11 12
                  0  1  0  1  0  0  0  0  1  1  0  0
```

```

      1001   Position 9
XOR 1010   Position 10
-----
      0011   Prüfbits berechnet
XOR 0110   Prüfbits empfangen
-----
=   0101   => Wert der Position 5 => Bit 5 ist falsch!
```

- Wurden 2 oder mehr Bits verändert, kann nur noch eine Aussage darüber getroffen werden, dass Bits verändert wurden. Die fehlerhaften Positionen können so nicht ermittelt werden.

1. Sie haben 8 Bit Nutzdaten (10011010) vorliegen.

- Ermitteln Sie die zu übertragenden Daten (Nutzdaten inklusive Prüfbits).

2. Überprüfen Sie, ob die folgenden Nachrichten korrekt übertragen wurden:

- 00111101
- 101110100010
- 001101100100
- 0001101100101101