

Versuch V7

C405 Hardwarepraktikum II

Abnahme: 20. Januar 2025

Stand: 4. Januar 2025

Tom Mohr

Martin Ohmeyer

Inhaltsverzeichnis

1	Kodierung	1
1.1	Notwendigkeit	1
1.2	Besondere Sequenzen	1
1.3	Datenblöcke	2
2	Implantation	3
2.1	Der Sender als Zustandsautomat	3
3	Verworfenene Ansätze	4

1 Kodierung

1.1 Notwendigkeit

Gesendete Nibble müssen auf der Leitung voneinander unterscheidbar sein. Um dies zu gewährleisten, wird ein Taktsignal in den Datenstrom kodiert, indem verhindert wird, dass ein gesendetes Nibble gleich seinem Vorgänger ist. Dies verursacht den Sonderfall zweier gleicher aufeinanderfolgender Nibble. Zu dessen Behandlung muss eine Escape-Sequenz definiert werden, welche die gleichen Nibble durchbricht. Durch die Einführung einer solchen Escape-Sequenz wird ihr eigenes Auftreten im Datenstrom jedoch selbst zu einem Sonderfall. Zur Handhabung dieser Sonderfälle und zur Definition von Blöcken wurde ein Protokoll zwischen Sender und Empfänger vereinbart.

1.2 Besondere Sequenzen

Aus den oben dargelegten Gründen werden vordefinierte Sequenzen in den Datenstrom injiziert. Eine solche Sequenz setzt sich aus vier festen (Escape-Sequenz) und vier dynamischen Bits (Handlungsanweisung: "Command") zusammen.

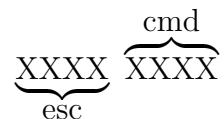


Abb. 1.1: Aufbau einer Kodierungssequenz

Die **Escape-Sequenz** trennt kodierende Sequenzen vom restlichen Datenstrom ab. Sie selbst hält keine Information darüber, um welche Kodierung es sich handelt. Aufgrund ihrer Sonderfunktion darf sie nicht regulär im Datenstrom auftreten und muss ggf. selbst escaped werden.

Commands erhalten erst dann ihre Bedeutung, wenn sie unmittelbar nach der Escape-Sequenz stehen. Sie geben Auskunft darüber, um welche Kodierung es sich handelt und implizieren, wie sich der Dekodierer verhalten muss, um die originalen Daten wieder zu rekonstruieren. Ist das nachfolgende Nibble auf einen Command (im binären) mit diesem identisch, wird anstelle des normalen Commands dessen Fallback-Version genutzt.

1 Kodierung

Hex	Bezeichnung	Bedeutung
0	escapeSequence	Das nächste nibble ist ein Command
1	beginDataBlockDefault	Ein Datenblock beginnt
2	beginDataBlockFallback	Ein Datenblock beginnt
3	beginControlBlockDefault	Ein Kontrollblock beginnt
4	beginControlBlockFallback	Ein Kontrollblock beginnt
5	endBlockDefault	Der aktuelle Block endet
a	endBlockFallback	Der aktuelle Block endet
6	insertPrevNibbleAgainDefault	Ein Nibble doppeltes im Datenstrom
7	insertPrevNibbleAgainFallback	Ein Nibble doppeltes im Datenstrom
8	insertEscSeqAsDataDefault	Die Esc-Seq trat im Datenstrom auf
9	insertEscSeqAsDataFallback	Die Esc-Seq trat im Datenstrom auf

Tabelle 1.1: Bitsequenzen und ihre Bedeutung

1.3 Datenblöcke

2 Implantation

2.1 Der Sender als Zustandsautomat

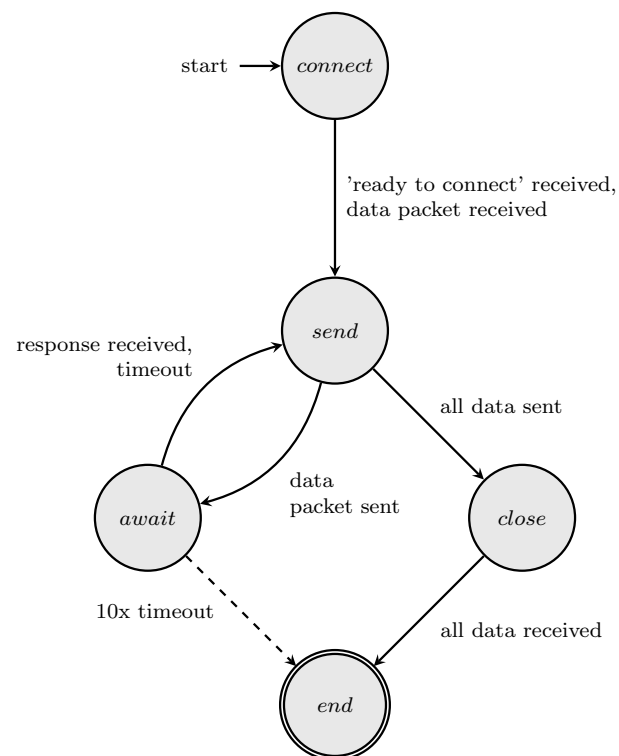


Abb. 2.1: Zustandsautomat des Senders

3 Verworfenene Ansätze