**Aufgabenblatt 10**

*Julian Bertol*

**Aufgabe 2**

Die erzeugte Foo.Wav Datei befindet sich im Projekt-Ordner.

Ausgabe hexer:  
Ein Bild, das Text, Screenshot, Majorelle Blue, Electric Blue (Farbe) enthält.

Automatisch generierte Beschreibung

Verständnisfrage:

Die Daten in der WAV-Datei werden sowohl auf dem Host (little-endian Architektur) als auch auf dem Target (big-endian Architektur) korrekt interpretiert, weil das WAV-Format standardmäßig little-endian ist. Das bedeutet, dass die Byte-Reihenfolge bei der Speicherung der Daten im little-endian Format erfolgt. Die meisten Audio-Bibliotheken und -Programme sind so implementiert, dass sie dies berücksichtigen und die Daten korrekt interpretieren können, unabhängig von der Endianess der zugrunde liegenden Hardware. Beim Erzeugen der PCM-Daten mit dem Format PCM\_16, also 16-Bit Integer, werden die Daten explizit in der little-endian Reihenfolge gespeichert, was eine korrekte Interpretation auf beiden Architekturen ermöglicht.

**Aufgabe 3:**

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

**Aufgabe 4:**Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

**Aufgabe 5**

Ein Bild, das Text, Screenshot, Software, Webseite enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Software, Webseite enthält.

Automatisch generierte Beschreibung

**Aufgabe 6**

Host:  
Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Target:

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Test:

Target:

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Host:

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Auf beiden Geräten wird ein Ton abgespielt

**Aufgabe 7**

Host:

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Target:

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Auch hier läuft auf beiden Geräten Musik

**Aufgabe 8**

gst-launch-1.0 filesrc location=hirsch.mp3 ! decodebin ! audioconvert ! tee name=t ! queue ! vorbisenc ! oggmux ! filesink location=cp.ogg t. ! queue ! audioresample ! alsasink

**Aufgabe 9**

Befehl auf dem Host

gst-launch-1.0 -v alsasrc ! audioconvert ! audioresample ! 'audio/x-raw,rate=16000,width=16,channels=1' ! speexenc ! rtpspeexpay ! udpsink host=<target-IP> port=6666

Befehl auf dem target:

gst-launch-1.0 udpsrc port=6666 caps="application/x-rtp,media=(string)audio,clock-rate=(int)16000,encoding-name=(string)SPEEX,payload=(int)110" ! rtpjitterbuffer ! rtpspeexdepay ! speexdec ! audioconvert ! audioresample ! alsasink

**Aufgabe 10**

Neuer Befehl:  
gst-launch-1.0 udpsrc port=6666 caps="application/x-rtp,media=(string)audio,clock-rate=(int)16000,encoding-name=(string)SPEEX,payload=(int)110" ! rtpjitterbuffer ! rtpspeexdepay ! speexdec ! tee name=t ! queue ! audioconvert ! audioresample ! alsasink t. ! queue ! audioconvert ! audioresample ! wavenc ! filesink location=recorded\_audio.wav