

Vorbereitung 10

Aufgabe 1, Android

b)

Wie funktioniert die CRC-Prüfung und welche Fehler können korrigiert werden?

Bei der zyklischen Redundanzprüfung wird aus vor der Übertragung der Bits ein CRC-Wert berechnet, um zu überprüfen, ob die Daten korrekt übertragen wurden.

Beispiel für ein CRC-Polynom:

$$p(x) = 1 \cdot x^n + 0 \cdot x^{n-1} + 0 \cdot x^{n-2} + \dots + 1 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0$$

Senderseite

1. Ein dem Sender und dem Empfänger bekanntes CRC-Polynom mit Grad n festlegen
2. An die zu übertragenden Daten n 0en anhängen (rechts)
3. Den erweiterten Datenrahmen mittels `xor` durch das CRC-Polynom dividieren
4. Den so berechneten Rest mit 0en auf Länge n ergänzen (links)
5. Datenrahmen versenden

Empfängerseite

1. Nachricht durch das CRC-Polynom dividieren
2. Je nach Rest der Division kann einer von 4 unten aufgelisteten Fällen auftreten

(Fehler)fälle

1. Der Rest der Division ist $= 0$ und die Nachricht ist richtig
2. Der Rest der Division ist $= 0$ und die Nachricht ist falsch. Dies kann passieren, wenn sowohl im Datenteil als auch im CRC-Rest ein Fehler passiert ist oder wenn das Fehlerpolynom ein Vielfaches des Generatorpolynoms ist. (eher unwahrscheinlich)
3. Der Rest der Division ist $\neq 0$ und die Nachricht ist falsch
4. Der Rest der Division ist $\neq 0$ und die Nachricht ist richtig. Dies kann passieren, wenn nur der CRC-Rest falsch empfangen wurde. (unwahrscheinlich, da Datenrahmen viel länger als CRC-Rest)

Erkenn-/Korrigierbare Fehler

Würde das CRC-Polynom gut gewählt ist es möglich verschiedene Arten von Fehlern zu beheben.

Einbitfehler Ist bei der Übertragung nur ein Bit gekippt nimmt das Fehlerpolynom die Form $F(x) = x^i$ an, wobei i der Index dieses Bits ist.

Zwei Einbitfehler Sind bei der Übertragung zwei (verschiedene) Bits gekippt nimmt das Fehlerpolynom die Form $F(x) = x^i + x^j$ an, wobei i und j die Indexe dieser Bits sind.

Ungerade Anzahl an Fehlern Mit einem CRC-Polynom das $(x + 1)$ als Faktor hat, ist es möglich alle Fehler mit ungerader Anzahl gekippter Bits zu erkennen.

Bündelfehler Solange die Länge des fehlerhaften Bereichs kürzer ist als der Grad des CRC-Polynoms ist es möglich solche Fehler zu erkennen.

c)

i.

Was für ein Protokoll kommt zum Einsatz? Wer ist für den Verbindungsaufbau zuständig?

Das Protokoll ist ein Master/Slave-Protokoll.
Master ist das Smartphone und Slave ist der FPGA.
Die Verbindung wird durch das Smartphone aufgebaut.

ii.

Wie groß ist die Paketgröße und warum wurde diese Größe gewählt?

Die Paketlänge ist auf genau 16 Bytes festgelegt. Eine feste Länge ist wichtig, da im FPGA im Voraus eine feste Anzahl an Registern für den Empfang eines Pakets reserviert werden muss.

Außerdem ist so ein einfacher Empfang im Mikrocontroller per DMA möglich.

iii.

Welche Übertragungsfehler können wie erkannt werden?

Als CRC-Polynom wurde $x^{16} + x^{12} + x^5 + 1$ gewählt.
Die erkennbaren Fehler wurden schon in **Teil b)** aufgelistet.

iv.

Welche Schritte sind für den Verbindungsaufbau und den Datenaustausch notwendig?

1. Bluetooth-Geräte suchen
2. (Verbindung aufbauen, wird an das nächste *Intent* übergeben)
3. (Datenübertragung durchführen, wird an das nächste *Intent* übergeben)

d)

i.

Alle aufgelisteten Schritte werden (auch) von der Klasse übernommen. Die dafür zuständigen Klassenmethoden sind:

- Die *onClick*-Methode des zu implementierenden *OnClickListeners* des *searchButtons* soll die Bluetooth-Geräte-Suche starten
- (Die *onItemClick*-Methode des zu implementierenden *OnItemClickListener* der *deviceList* soll die Verbindung zum Bluetooth-Gerät aufbauen bzw. das ausgewählte Bluetooth-Gerät an das nächste *Intent* weitergeben)
- (Die eigentliche Datenübertragung erfolgt in Anroid über einen *BluetoothSocket*)

iv.

Der *BroadcastReceiver* hat die Aufgabe Broadcast-Daten zu verarbeiten. Dazu wird eine *onReceive*-Methode implementiert. in unserem konkreten Fall werden die Nachrichten *ACTION_FOUND*, *ACTION_DISCOVERY_STARTED* und *ACTION_DISCOVERY_FINISHED* der Bluetooth-Geräte-Suche verarbeitet.

Bei *ACTION_FOUND* wird das gefundene Gerät der *deviceList* hinzugefügt.

Bei *ACTION_DISCOVERY_STARTED* wird die Progress-Bar angezeigt und bei *ACTION_DISCOVERY_FINISHED* entsprechend wieder versteckt. Weiterhin werden zu jeder dieser Nachrichten Log-Nachrichten sowie Toast-Nachrichten abgesetzt.

Aufgabe 2, FPGA

a)

- Aktor: 8x binär, LEDs
- Sensor: 4x binär, Buttons
- Sensor: 8x binär, Switches
- Aktor: 1x 7-Segment Anzeige

- Sensor: 8x 12-Bit Analogsensoren
- Sensor: 1x binär, SensorDoor
- Aktor: 4x PWM
- Aktor: 2x binär, PeltierDirection und BTHouseOn
- Regler und Zielwert: 2x, Temperatur und Licht

b)

Dort ist ein Stift bei der Aufgabe, aber es gibt nichts zu schreiben.

c)

Siehe **Aufgabe 1 Teil b)**.