


10 Versuchsreihe

Diese abschließende Versuchsreihe des Hardwarepraktikums befasst sich mit der Kommunikation zwischen FPGA und dem Android-Smartphone.

Kompetenzen Die Studierenden

- können Schaltungen mittels einer Beschreibungssprache entwerfen,
- verstehen den Umgang mit der Hardwarebeschreibungssprache VHDL und deren Zusammenspiel mit der Hardware,
- zeigen Lernbereitschaft und besitzen Offenheit für neue Ideen und Anforderungen,
- kennen die besonderen Randbedingungen des Entwurfs eingebetteter Systeme,
- verstehen den Zusammenhang zwischen Hardware-Konzepten und den Auswirkungen auf die Software bzw. die Anwendung,
- können im beruflichen Umfeld mit Ingenieuren und Elektrotechnikern als Informatikanwender kommunizieren.

10.1 Vorbereitung

Hinweis: Die Lösungen zu den Vorbereitungsaufgaben, insbesondere jene die mit einem  gekennzeichnet sind, müssen bis **spätestens 18 Uhr am Vortag des Laborterminals** per Sciebo (einmal pro Gruppe, Bsp.: für die Montagsgruppe wäre dies Sonntag 18 Uhr) über folgenden Link hochgeladen werden:

Für die Gruppen von Montag (Passwort: Transistor):

<https://uni-siegen.sciebo.de/s/g4Nv14QVgFzk71H/authenticate>

Für die Gruppen von Dienstag (Passwort: Diode):

<https://uni-siegen.sciebo.de/s/cdfkb0cHMTvsmEk/authenticate>

Für die Gruppen von Mittwoch (Passwort: Kondensator):

<https://uni-siegen.sciebo.de/s/64Lje097Gdaw4mB/authenticate>

Dateiname: Gruppenname_ Versuchsreihe













Beispiel: Gruppe 01 und Vorbereitung für Versuch 1: G01_ V1






Wenn der Link unter Sciebo temporär nicht zu erreichen ist, dürfen Sie die Vorbereitung auch an folgende Mail-Adresse verschicken: ferid.mahdi@uni-siegen.de.

Fehlendes Zusenden der Lösungen führt zum Ausschluss vom Labortermin!

1. **Android:** Um das Modellhaus extern steuern zu können, wurde eine Android-App entwickelt. Für die Kommunikation über Bluetooth müssen noch einige Angaben in der App ergänzt werden. Bearbeiten Sie hierfür in der Vorbereitung noch folgende





Aufgaben:

- a)  Lesen Sie sich mit Hilfe des Anhangs in die technischen Aspekte der Bluetooth-Spezifikation ein.
 - b)   Um Fehler bei der Übertragung zu vermeiden oder zu korrigieren, existieren unterschiedliche Verfahren. Wie funktioniert die CRC-Prüfung und welche Fehler können korrigiert werden?
 - c)  Lesen Sie im Projekthandbuch der PG ASHA nach, wie die PG die Bluetooth-Verbindung und das Protokoll zwischen FPGA und Android-Smartphone realisiert hat und beantworten Sie folgende Fragen:
 - i.  Was für ein Protokoll kommt zum Einsatz? Wer ist für den Verbindungsaufbau zuständig?
 - ii.  Wie groß ist die Paketgröße und warum wurde diese Größe gewählt?
 - iii.  Welche Schritte sind für den Verbindungsaufbau und den Datenaustausch notwendig?
 - d)  Laden Sie sich die Datei *AshaAppJava.zip* von Moodle auf Ihren Computer und importieren Sie das Projekt in Android Studio. Lesen Sie hierzu das Kapitel 4 des Handbuchs der Projektgruppe ASHA. Die App wurde Anfang 2020 aktualisiert und ist mit Android 10 getestet worden. Durch die Aktualisierung haben sich die Oberflächen leicht geändert. So wurde die *MainActivity* über eine Schaltfläche erweitert, welche die Bluetoothgeräte filtert und nur die Asha-Modellhäuser (inklusive der entsprechenden IDs) anzeigt. Durch die Oberflächen-Umstrukturierung von Android sind die beiden Oberflächen (Details und History) eines Sensors durch ein neues Design-Konzept ersetzt worden (Fragments). Ebenso ist das Einstellungs-Menü über die drei Punkte in der oberen Menüleiste erreichbar und die Sprache kann zwischen Deutsch und Englisch geändert werden. Lösen Sie folgende Aufgaben (das in Versuch 9 bereitgestellte Tutorial kann ebenfalls als Hilfe dienen):
 - i.  Schauen Sie sich die Klasse *MainActivity* der gegebenen Android-Anwendung im Detail an. Welche der zuvor aufgelisteten Schritte (siehe 1.c.iii) werden von dieser Klasse übernommen? Welche Methoden sind dafür verantwortlich?
 - ii.  Durchsuchen Sie die Klasse *MainActivity* nach *TODO-1* und *TODO-2* und ergänzen Sie die fehlenden Codezeilen.
 - iii.  Implementieren Sie die Methode *setOnItemClickListener* (*TODO-3*) der Liste *listDevices*.
 - iv.  Welche Aufgabe hat der *BroadcastReceiver*? Welche Nachrichten werden von ihm verarbeitet und was passiert im Einzelnen, wenn diese Nachrichten eintreffen?
2. **FPGA:** Um mit dem FPGA eine Verbindung zu einem Android-Gerät aufbauen zu können, werden die Module Bluetooth, UART und CRC benötigt. Bearbeiten Sie hierzu folgende Aufgaben.

- a)   Schauen Sie sich den umfangreichen Prozess **Bluetooth** im Modul **Bluetooth** an. Für welche Sensoren und Aktoren werden hier Daten versendet? Listen Sie alle Sensoren und Aktoren der Reihenfolge nach auf.
- b)   Neben den Protokolleinstellungen und dem Senden und Empfangen der Pakete stellt die Aufbereitung der Daten eine wichtige Rolle da. In dem **Bluetooth**-Prozess ist die *Sensorverarbeitung* bereits implementiert. Ergänzen Sie an der markierten Stelle die *Aktorverarbeitung*, in dem Sie sich an der Sensorverarbeitung orientieren. Verarbeitet werden sollen Daten der 6 LEDs, der 7-Segmentanzeige, der beiden PWM-Lüfter, des PWM-Lichts, des PWM-Peltiers, der Peltier-Richtung, der Haussteuerung und den Sollwerten für Temperatur und Helligkeit.
- c)  Beantworten Sie die Fragen des 10. Moodle-Selbsttestes. Sollten Sie bei der Beantwortung Probleme haben, müssen Sie ggf. Teile der Vorbereitung nochmal wiederholen.

Tipp: Schauen Sie sich am besten auch schon die Aufgaben der Durchführung an und führen Sie gegebenenfalls selbstständig weitere Vorbereitungen durch, damit Sie in der Laborphase nicht unnötig Zeit verlieren!

10.2 Durchführung

1.   Realisieren Sie die oben beschriebenen Änderungen für die Bluetooth-Kommunikation mit dem FPGA. In Abbildung 10.1 ist die bereits aus den Versuchen 6, 7 und 9 bekannte FPGA Schalterbelegung dargestellt. Implementieren Sie den *Modus 0: Steuerung über die Smartphone App*, indem Sie die markierte Stelle im Modul *actor* ergänzen. Auf Grund der Empfindlichkeit der Bluetooth-Verbindung in Verbindung mit den auftretenden Strömen des Peltier-Elements, geben Sie am besten für das Peltier-Element einen festen Wert vor (also für *PWM4PeltierValue* und *PeltierDirection*), und übernehmen nur für die LEDs, die Siebensegmentanzeige, die Lüfter und das Hauslicht die Bluetoothwerte. Testen Sie einmal das Gesamtsystem mit der Bluetooth Anbindung zu einem Android-Gerät. Sollten Sie kein Android-Gerät besitzen, wird ihnen hierfür eins vom Betreuer geliehen.
Hinweis: Die MAC-Adresse ihres Bluetooth-Moduls befindet sich auf einem Aufkleber auf dem Gerät, der Pin zur Kopplung mit der App lautet „1234“.
2.   Lassen Sie sich die gesamte Funktionalität des Hauses (Sensorwerte auslesen, Aktoren in Betrieb nehmen, Regelungsmodus und Steuerung über Bluetooth) vom Betreuer abzeichnen.

FPGA Switch und Button Belegung

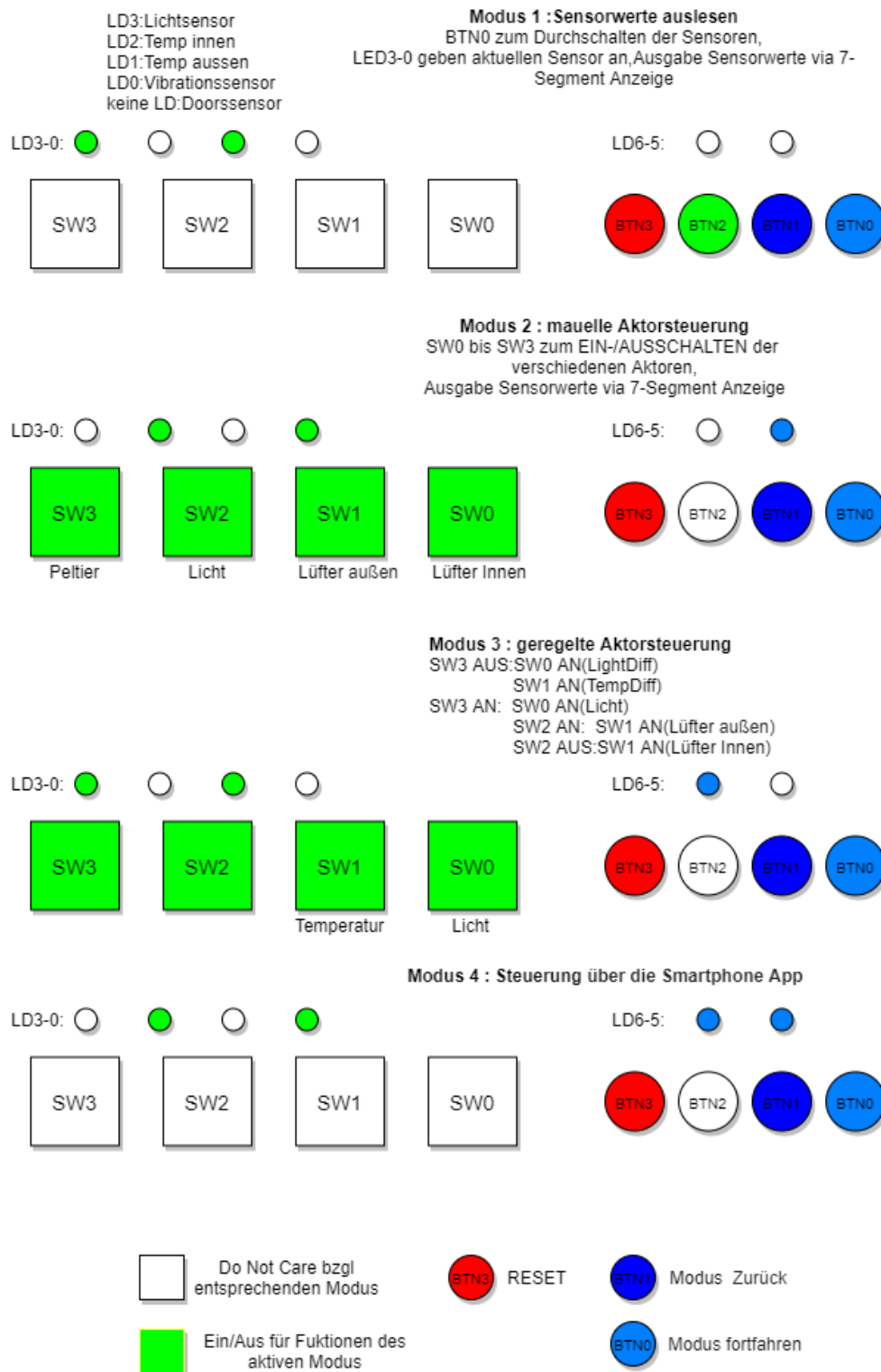


Abbildung 10.1: FPGA-Ansteuerung des Hauses

10.3 Protokollierung

1. ✎ Schreiben Sie einen kurzen Text in dem Sie beschreiben, welche Module im Projekt für welche der folgenden Aufgaben zuständig sind:

- Auslesen der Sensorwerte und Darstellung auf der Siebensegment-Anzeige
- Ansteuerung der Aktoren
- Automatische Regelung der Aktoren
- Kommunikation über Bluetooth

Welche Aufgaben sind Ihnen bei der Haussteuerung am leichtesten, welche am schwersten gefallen?

2. ✎ Fügen Sie außerdem Ihre (ggf. korrigierten) Notizen aus der Vorbereitung dem Protokoll hinzu!
3. 💻 Packen Sie Ihr Projekt mit dem gesamten Quellcode als Zip-Datei. Laden Sie den Quellcode (ZIP) einmal pro Gruppe bis zum 03.07.2019, 16.00 Uhr über Ihren Sciebo-Link hoch.

10.4 Anhang

10.4.1 Grundlagen Bluetooth

10.4.1.1 Allgemeines

Bluetooth wurde ab 1990 hauptsächlich von Ericsson und Nokia entwickelt. Benannt wurde die Technologie nach dem dänischen Wikingerkönig Harald Blauzahn, der für seine ausgezeichneten Kommunikationsfähigkeiten bekannt war.

Geräte nach den Standards der Bluetooth SIG senden als Short Range Devices (SRD) in einem lizenzfreien ISM-Band (Industrial, Scientific and Medical Band) zwischen 2,402 GHz und 2,480 GHz. Sie dürfen weltweit zulassungsfrei betrieben werden. Störungen können aber zum Beispiel durch WLANs, Schnurlostelefone oder Mikrowellenherde verursacht werden, die im gleichen Frequenzband arbeiten. Um Robustheit gegenüber Störungen zu erreichen, wird ein Frequenzsprungverfahren (Frequency Hopping) eingesetzt, bei dem das Frequenzband in 79 verschiedene Frequenzstufen im 1-MHz-Abstand eingeteilt wird, die bis zu 1.600-mal in der Sekunde gewechselt werden. Es gibt jedoch auch Pakettypen, bei denen die Frequenz nicht so oft gewechselt wird (Multislot-Pakete) [...]

Ab der Version 2.0 + EDR können Daten mit EDR (Enhanced Data Rate) maximal etwa dreimal so schnell wie bisher übertragen werden, also mit rund 2,1 Mbit/s. Bereits ab Version 1.1 kann ein Bluetooth-Gerät gleichzeitig bis zu sieben Verbindungen aufrechterhalten, wobei sich die beteiligten Geräte die verfügbare Bandbreite teilen müssen (shared medium).

Bluetooth unterstützt die Übertragung von Sprache und Daten. Allerdings können die meisten Geräte während der Übertragung von Sprache aufgrund der synchronen Verbindung lediglich drei Teilnehmer in einem Piconet verwalten.

Eine Sicherung und eine Verschlüsselung der transportierten Daten sind ebenfalls möglich. (Quelle: Wikipedia)

Teile der Spezifikation bzw. der Architektur können auf der Projektwebsite eingesehen werden ¹.

10.4.1.2 Reichweite

Klasse	Max. Leistung (Watt)	Reichweite im Freien
Klasse 1	100 mW	ca. 100 m
Klasse 2	2,5 mW	ca. 50 m
Klasse 3	1 mW	ca. 10 m

Tabelle 10.1: Bluetooth Klassen: Reichweite (Quelle: Wikipedia)

¹<https://www.bluetooth.org/Building/HowTechnologyWorks/Architecture/Overview.htm>

10.4.1.3 Testbarkeit

Obwohl mit dem Android SDK die meisten Funktionalitäten eines modernen Smartphones emuliert werden können, gilt dies jedoch nicht für alle. Insbesondere eine Kommunikation über Bluetooth lässt sich derzeit noch nicht testen. Um die Korrektheit der Übertragung zu überprüfen, bleibt Ihnen nur die Möglichkeit, Ihre Implementierung mit Hilfe des kompletten Aufbaus im Labor zu testen.

10.4.2 Android

10.4.2.1 Aufgabe: AshaApp

Um den Einstieg in die Bluetoothprogrammierung zu erleichtern, wird ein Teil der Funktionalität bereits vorgegeben. Im Moodlekurs finden Sie ein Archiv mit den entsprechenden Dokumenten. In der nachstehenden Tabelle [10.2](#) sind die Klassen und deren Funktionen beschrieben. Wenn sie das Projekt geöffnet haben, werden Sie einige Warnungen sehen. Dies ist nicht weiter schlimm, da die IDE sehr gut kontrolliert und nicht benutzte Methoden oder mit zu hoher Sichtweite, obwohl niedrigere ausreichend wäre, mit einer Warnung anzeigt. Die App wurde minimal modifiziert und entspricht dadurch nicht mehr ganz dem Projekthandbuch, so wurden beispielsweise die Layouts angepasst und einen Schalter zum filtern der Geräte integriert. Wenn ein Asha-Gerät erkannt wurde, wird ebenfalls die numerische ID in der Liste angezeigt.

Klassenname - ui	Aufgabe
MainActivity	Die grundlegende Interaktionsfläche der Applikation
DeviceListActivity	Zeigt die aktuellen Werte eines Gerätes an.
LoggerActivity	Zeigt die Logs an
AktorActivity	Zeigt die Details eines Aktors an, der Aktor kann gesteuert werden
SensorFragmentActivity	Zeigt Details und History eines Sensors durch zwei Fragmente an
DeviceFragment	Sensor Details Oberflächenfragment
HistoryFragment	Sensor History Oberflächenfragment
HistoryView	Grafisches Verlaufsdiagramm
Klassenname - kernel	Aufgabe
Bluetooth	Stellt Basisfunktionalität der Bluetoothkommunikation bereit
CRC16	Klasse zum Berechnen von CRC16
Device	Generische Klasse für einen Sensor oder Aktor
DeviceAdapter	Erweitert die Klasse ArrayAdapter um Funktionen für die Aktoren und Sensoren
.DeviceHolder	Speichert die TextView Referenzen
DeviceComparator	Comparator Klasse zum Vergleichen von zwei Devices
DeviceContainer	Container für ausgelesene Sensoren/Aktoren
Logger	Verwaltung des Logging
Protocol	Beinhaltet Vereinbarungen bzgl. des Protokolls
.DeviceType (internal)	Art des Gerätes
.DeviceSubType1	Art des Sensors
.DeviceSubType2	Art der Ansteuerung des Gerätes
.DataType	Typ in der Rückgabewert interpretiert werden soll
Utils	Enthält Funktionen, um die Sprache zu ändern

Tabelle 10.2: Android: Wichtige Klassen und Funktionen