

Praktikum DT im WiSe2012/13 am Mi 15:15

Prof. Dr. D. Jantz

Vorgehen im Projekt:

Das Projekt soll im Team realisiert werden. Gruppenstärke 3 bis 5 Leute. Dazu stehen 7 Termine zur Verfügung (1 Vorbesprechung, 4 Arbeitstermine, 1 Reservetermin, 1 Präsentationstermin).

Beim letzten Arbeitstermin oder spätestens zum Reservetermin erfolgt eine Abnahme durch die Geschäftsleitung (Betreuer). Diese Abnahme ist Grundvoraussetzung für die Zulassung zur Präsentation und zur Prüfung. Wägen Sie den Aufwand also ab, zuwenig kostet ein Semester, zu viel bringt nichts.

Beim Präsentationstermin hält die Gruppe einen Vortrag. Jeder Teilnehmer präsentiert seinen Teil. Der Gruppenleiter führt durch den Vortrag. Es erfolgt eine Beurteilung der Gruppe.

Am Ende des Praktikums schließt sich eine mündliche oder schriftliche Prüfung an. Hier „kämpft“ jeder für sich! Es gibt eine Einzelnote.

Das Projekt gliedert sich in (min.) 7 Teile :

1. Konzepterstellung, Schnittstellen, Teilprojekte, Aufteilung
2. Kurzvortrag der Planung und Konzeption, im Labor / Beamer
3. Entwicklung der Teilprojekte, Simulation (jeder Teil hat eigenen Rechner)
4. Zusammenbau der Teilprojekte
5. Hardwareabschlusstest (Beweis der Funktionalität)
6. Präsentation des Projektes durch die Gruppe
7. Prüfung / Klausur

Wählen Sie aus Ihrer Mitte einen Projektleiter.

Der Projektleiter ist für die kontrollierte Abwicklung des Projekts verantwortlich. Er teilt die Aufgaben auf, koordiniert die Schnittstellen, sorgt für die nötige Ausrüstung (Oszilloskop, Funktionsgenerator, zusätzliche Chips etc.).

Die einzelnen Teams bearbeiten und testen Ihre Module selbständig nach vorgegebenem Zeitplan des Projektleiters. Die Einhaltung der Schnittstellen und des Zeitplans ist obligatorisch. Abweichungen werden mit der Geschäftsleitung (Betreuer) besprochen.

Anwesenheitspflicht besteht für Vorbesprechung, Konzeptbesprechung, Abnahme, Präsentation und Prüfung.

Die Endnote setzt sich aus der Präsentationsnote und einer schriftlichen Prüfung (oder einem mündlichen Abschlusskolloquium) zusammen.

Viel Spaß und Erfolg

P.S. Man kann sich durch das Praktikum mogeln. Sie lernen aber dann nichts, die Zeit ist verschwendet und die Prüfung zeigt es auf!

Vorbereitung:

Machen Sie sich mit dem Altium LiveDesign Board (Evaluation Board, kurz Eval Board) vertraut, studieren Sie den Schaltplan und die Beschreibung, holen Sie sich das Datenblatt des Bausteins und testen Sie die Beispiele "counter_simulation" (Simulation) und "counter_7seg_board" (Implementieren) mit der Xilinx Software ISE Webpack.

Speziell für die Tastatur gibt es das Beispiel „Tastatur“ das mit dem Konfigurationsfile rapidIO_Tastatur_Flachband.LPF für den LogiPort Logikanalyzer zusammenspielt.

Stellen Sie sich Fragen aus der Vorlesung oder der Prüfung DT (1), die Sie noch im Gedächtnis haben. Tauschen Sie die Fragen innerhalb der Gruppe aus. Wiederholen Sie die entsprechenden Teile der Vorlesung

Hinweise:

Achtung: der Xilinx Baustein „verträgt“ nur 3,3V Inputs (max. 4,2 V), Vorsicht beim Anschluss externer Hardware !!!

Projekt-Aufgabe VGA / Videospiele (VGA)

- Bildschirmgrundfarbe (simple VGA digital-kompatibel, keine 24bit Farbsteuerung nötig)
- mehrere Taster auf Board zur Steuerung
- Min. ein andersfarbiges Rechteck von $1/4h \times 1/4b$ auf dem Bildschirm bewegen können
- 7seg-Display zur Kontrolle
- relativ flinke Echtzeitsache
- Dokumentation VGA suchen!

Mit dem Eval Board soll ein VGA Monitor soll angesteuert werden und eine selbst zu definierende Grafik dargestellt werden (kann auch ein Videospiele werden). Simulieren Sie (bedarfsweise) alle Teilaufgaben.

Der VGA Monitor wird über das Eval Board digital angesteuert. Für jede Grundfarbe (RGB) existieren 3 Ausgänge, die zu einem Eingang verschaltet sind (Siehe Beschreibung der IO's).

Außer den 3 Signalen RGB (rot, grün, blau auf VGA Buchse 1, 2, 3) benötigt der Monitor 2 Synchronisationspulse für die horizontale und vertikale Ablenkung (Horizontaler Synchronisationspuls auf VGA Stecker 13, vertikaler Synchronisationspuls auf VGA Stecker 14). Die Synchronisationspulse müssen mindestens 3 - 4 μ s lang und low aktiv sein.

Ein VGA Monitor arbeitet mit einer Horizontalfrequenz $f_H = 31,25$ kHz und einer Vertikalfrequenz $f_V = 60$ Hz. Bei einer Taktfrequenz von 50 MHz kommt man damit gut hin.

Konkretisieren Sie Ihre eigene Aufgabenstellung weiter und halten diese schriftlich fest!

Projekt-Aufgabe Funkuhr (FUh)

=>Variante Funkuhr-Generator „Sender“

- Auslesen des DCF-77 Signal
- Darstellen von Uhrzeit hhhmss
- Auf Tasterdruck umschalten zu Datum ddmmyy
mit Board-LED anzeigen welcher Modus
- langsam aber Echtzeitsache, messen, nur 1x Hardware

Bauen Sie mit Hilfe des Eval Boards ein Funkuhrmodul auf.

Simulieren Sie (bedarfsweise) alle Teilaufgaben.

Der Analogteil (Empfänger von Conrad, beim Betreuer nachfragen) steht zur Verfügung. Es handelt sich um ein Modul, das die Funkwellen aus Frankfurt bereits in ein digitales Signal umwandelt. Das Digitalsignal liegt als Open Kollektor npn positiv und negativ oder sogar logisch über einen Puffer vor. Untersuchen Sie das Signal mit dem Oszilloskop.

Informationen unter <http://www.heret.de/funkuhr/ptb.htm>

Konkretisieren Sie Ihre eigene Aufgabenstellung weiter und halten diese schriftlich fest!

Projekt-Aufgabe Tastatur / Taschenrechner (TRe)

- Rechnen in Dezimal oder Hexadezimal
- 7seg-Display mit +/- Vorzeichen und 5 Stellen verwenden
- Bedienung wie Taschenrechner, z.B. $10+12=22$, $-1+8=7$
- Funktionen (5*): ADD $y+x$, SUB $y-x$, AND $y&x$, OR $y|x$, XOR $y<=>x$,
- LED-Anzeige für Status: 1.Operand, 2., Ergebnis
- Erkennen eines ERRORS, Anzeige ERROR (R=A)
- langsam, aber knifflig
- Dokumentation Tastatur suchen!

Informationen über eine PC Tastatur finden Sie im Anhang oder über die Praktikumsseite des Labors Industrielle Elektronik

Konkretisieren Sie Ihre eigene Aufgabenstellung weiter und halten diese schriftlich fest!

Projekt-Aufgabe Tastatur / Spiel (ff.) (TSp)

- Anzeigemöglichkeiten: 6 Stellen auf 7seg-Display, einige LEDs
- Steuerung per Tastatur, evtl. Umschaltung Spiele/Spielmodi
- Ideen ala Lunar-Lander, Bediener soll gegen etwas gegensteuern
- kreativ auszufüllen, eigene Gestaltung
- Anschluss zusätzlicher Anzeigehardware (LED z.B.) ist möglich
- Dokumentation Tastatur und WAV-Sound suchen!



Informationen über eine PC Tastatur finden Sie im Anhang oder über die Praktikumsseite des Labors Industrielle Elektronik

Konkretisieren Sie Ihre eigene Aufgabenstellung weiter und halten diese schriftlich fest!


Anhang: Informationen über PC Tastatur

Die Anschlussbelegung

Ältere AT Tastaturen haben einen 5poligen DIN Stecker am Kabelende, modernere Tastaturen besitzen jedoch einen PS2 Stecker, die neuen einen USB-Stecker, den wir hier aber nicht gebrauchen können.

Stecker 5polig DIN:  Vorderansicht	Pin 1: KBClock Pin 2: KBData Pin 3: n.c. Pin 4: GND Pin 5: + 5V	Stecker PS2:  Vorderansicht	Pin 1: KBClock→P1.0 (ge) Pin 2: GND (blau) Pin 3: KBData→P1.1 (gr) Pin 4: n.c. Pin 5: +5V (or) Pin 6: n.c.
--	--	--	--

Tab.1 : Pinbelegung von Tastaturstecker

Buchse PS2:  Vorderansicht	Pin 6: KBClock→P1.0 (ge) Pin 5: GND (blau) Pin 4: KBData→P1.1 (gr) Pin 3: n.c. Pin 2: +5V (or) Pin 1: n.c.
--	--

Tab.2 : Pinbelegung PS2 Buchse

Die Datenübertragung

Die Datenübertragung einer Tastatur geschieht seriell und ist bidirektional möglich, d.h. die Tastatur kann auch Daten empfangen. Zur Synchronisation zwischen Tastatur und angeschlossenem Gerät erfolgt über eine zusätzliche Leitung, die CLOCK-Leitung.

Die Richtung zur Tastatur hat die höhere Priorität, d.h. es kann jederzeit (auch wenn die Tastatur gerade sendet) ein Befehl an die Tastatur gesendet werden.

* Protokoll falls Tastatur = Sender

Das von der Tastatur erzeugte CLOCK-Signal hat eine typische Frequenz von 20 bis 30 kHz.

Die Tastatur kann nur Daten senden wenn CLOCK und DATA high (=Idle) sind.

Startbit : 1 (Pegel: Low)

Datenbits : 8

Paritybit : 1 (Ungerade Parität)

Stopbit : 1 (Pegel: High)

Bei jeder fallenden Flanke des CLOCK-Signals kann ein Bit des DATA-Signals eingelesen werden.

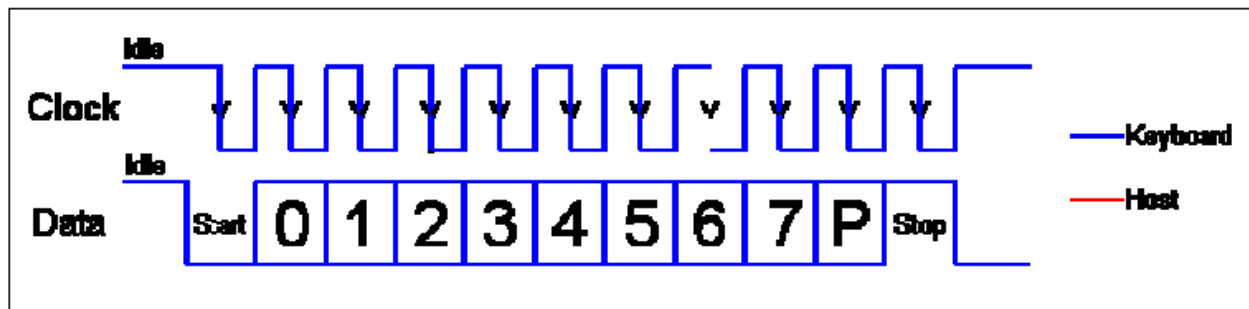


Abb.1 : Übertragungsprotokoll Tastatur = Sender

* Protokoll falls Tastatur = Empfänger

Dieses Protokoll wird verwendet um SteuerCodes zur Tastatur zu senden. Zu diesen SteuerCodes gehören zum Beispiel:

- Aktivieren der CAPS LOCK - LED (Hochstelltaste)
- Aktivieren der NUM - LED

Der Übertragung wird dadurch eingeleitet, dass die DATA - Leitung auf Low gezogen wird. Um zu verhindern, dass zur selben Zeit Daten von der Tastatur geschickt werden, muß die CLOCK - Leitung für 60µs auf Low gezogen werden. Dann erst wird die DATA - Leitung Low und die CLOCK - Leitung wieder der Tastatur überlassen. Nun folgen angefangen mit dem LSB (Least Significant Bit) die 8 Datenbits mit dem Paritybit. Nachdem das Paritätsbit gesendet wurde und die DATA Leitung für eine weitere CLK Periode wieder im High Zustand ist, wird die Tastatur eine Bestätigung(ACK) für den Empfang der neuen Daten senden. Die Tastatur tut dies, indem sie die DATA Leitung für die nächste CLK Flanke auf Low zieht. Ist die DATA Leitung nach dem 10ten Bit (Start+8Data+Par.) nicht High („idle“), sendet die Tastatur weiterhin das CLK Signal bis die DATA Leitung High („idle“) wird.

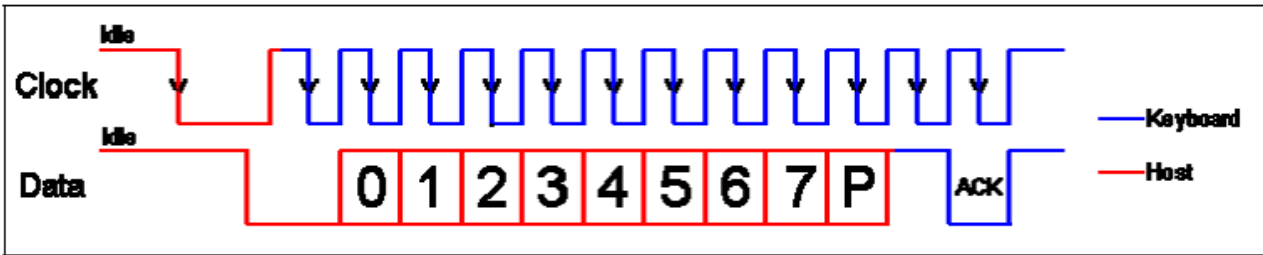


Abb.2 : Übertragungsprotokoll Tastatur = Empfänger

Der Tastatur-Code

Beim Bedienen der Tastatur wird nicht nur ein „Zeichen“ pro gedrückte Taste übertragen, sondern eine ganze Folge von Zeichen. Bei der Benutzung der Tastatur kommen verschiedene Zustände vor. Eine Taste kann kurz oder lang gedrückt werden oder mehrere Tasten werden gleichzeitig betätigt. Tastaturen werden in verschiedenen Ländern benötigt, diese haben unterschiedliche Schriftzeichen. Durch den verwendeten Code ist nur eine Änderung der Tastenaufschrift nötig und keine Hardwareänderungen. Jede Taste der Tastatur hat einen bestimmten Code, den Scancode, der gesendet wird.

Scan-Code

Diese Scancodes müssen vom Empfänger ausgewertet werden. Jeder Scancode ist ein Byte lang. Durch das Drücken der Taste ‚A‘ zum Beispiel, wird der Scancode 1C(hex) von der Tastatur erzeugt. Wenn man die Taste nicht loslässt, wird nach einer bestimmten Verzögerung wieder 1C (hex) gesendet. Das wiederholt sich, bis die Taste losgelassen wird oder eine andere Taste gedrückt wird.

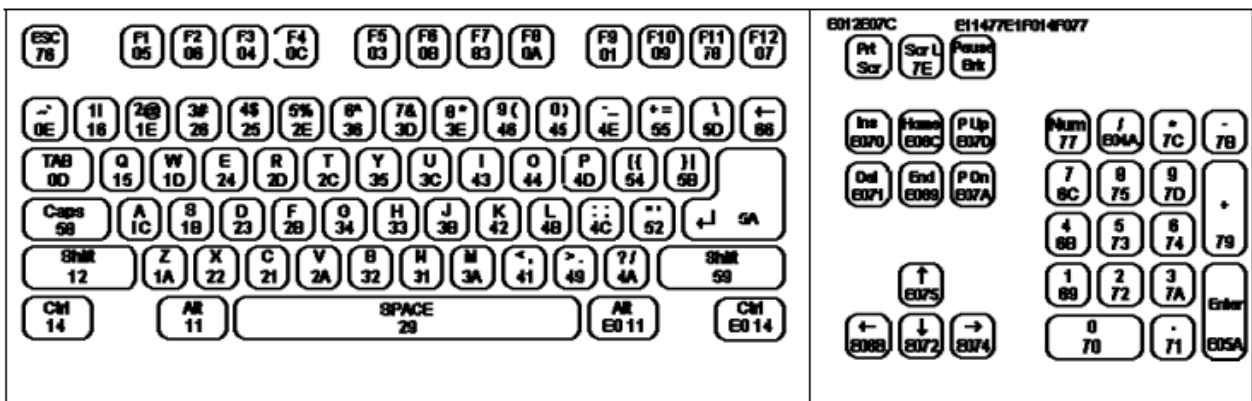


Abb.3 : Scan-Codes mit amerikanischer Beschriftung
Tastenbeschriftung

Release-Code

Zusätzlich sendet die Tastatur beim Loslassen der Taste noch weiteren Code, bestehend aus zwei nacheinander gesendeten Bytes.

Bei der Taste ‚A‘ ist das zum Beispiel:

Release-Code (Code für Loslassen)	Scancode der gerade losgelassenen Taste
F0 _{hex}	1C _{hex}

Tab.3 : Loslassen einer Taste

Beispiele zum erzeugten Tastaturcode

Es folgen zwei Beispiele für den gesendeten Tastaturcode. Das erste Beispiel behandelt das Drücken und Loslassen einer einzelnen Taste.

Das zweite Beispiel behandelt die etwas kompliziertere Tastenkombination.

Einzelne Taste

Aktion des Benutzers	Tastaturcode	Erklärung
Taste „a“ drücken und gedrückt halten	0x1C..0x1C..0x1C... C...	Scancode x-mal senden (1x mindestens)
Taste „a“ wieder loslassen	0xF0	Releasecode senden
	0x1C	Scancode der losgelassenen Taste 1x senden

Tab.4 : Ablauf beim Drücken einer einzelnen Taste

Tastenkombination

Die Tastatur hat nur einen Scancode pro Taste. Es ist egal ob zusätzlich die SHIFT-Taste gedrückt wurde, es wird trotzdem derselbe Code gesendet.

Aktion des Benutzers	Tastaturcode	Erklärung
Linke SHIFT drücken und halten	0x12 0x12 0x12 ...	x-mal senden bis zusätzliche Taste gedrückt (SHIFT wird nicht losgelassen, d.h. kein Release-Code 0xF0)

'a' drücken und halten	0x1C 0x1C 0x1C ...	x-mal senden
Linke SHIFT loslassen	0xF0	Senden des Releasecodes 0xF0
	0x12	Scancode der Taste die losgelassen wird
	(0x1C ...)	da ,a' noch gedrückt, wird weiterhin 0x1c gesendet
,a' loslassen	0xF0	Senden des Releasecodes 0xF0
	0x1C	Scancode der Taste die losgelassen wird

Tab.5 : Ablauf beim Drücken einer Tastenkombination

kurz: 0x12 0x1c 0x1c 0x1c 0xF0 0x12 (0x1c) 0xf0 0x1c

Anmerkung: der eingeklammerte Scancode 0x1c (,a') kommt nicht wenn ,a' mit Shift gleichzeitig losgelassen werden!