

HAW Hamburg - Prüfungsklausur Computertechnik - WS 2008/09					
Aufgabe	1	2	3	4	Summe
Punkte	12	25	47	16	100
	15	25	30	54	100

1 Adressbereiche

An einen Controller H8S/2357 sollen drei Speicherchips angeschlossen werden:

- Der erste Speicherchip ist ein ROM-Chip mit 2 Megabyte Kapazität. Der Adressbereich des ROM-Chip soll an den unteren Adressen des Controllers ab 0x000000 liegen.
- Der zweite Speicherchip ist ein EEPROM-Chip mit 2 Megabyte Kapazität. Der Adressbereich des EEPROM-Chip soll ab der Controller-Adresse 0x800000 beginnen.
- Der dritte Speicherchip ist ein RAM-Chip mit 2 Megabyte Kapazität. Der Adressbereich dieses RAM-Chip umfasst die obersten Adressen des Controllers und endet mit der Adresse 0xFFFFF.

a) Welche Adressbereiche bleiben frei für spätere Erweiterungen? Nennen Sie die freie Kapazität in MB, die Anfangs- und Endadresse (hexadezimal).

Kapazität	Anfangsadresse	Endadresse
6 MB	0x02	0x07
4 MB	0x0A	0x0D
2 MB	0x0E	0x0F
4 MB	0x0A	0x0B
2 MB	0x0C	0x0D
2 MB	0x0E	0x0F

b) Geben Sie die Gleichungen der Chipselect-Signale (\overline{CS}) von ROM, EEPROM und RAM an, wenn diese aus den Adressleitungen des Controllers erzeugt werden.

ROM: $\overline{CS} = A_{15} + A_{12} + A_{11}$

EEPROM: $\overline{CS} = \overline{A_{15}} + A_{12} + A_{11}$

RAM: $\overline{CS} = \overline{A_{15}} + \overline{A_{12}} + \overline{A_{11}}$

SS/WS	Semester	Fach	Datum
09	4	CT	2008

FSR: Klausurenabteilung 13

2 Periodische Signalfolgen mit der TPU erzeugen

Die folgenden Signale (Bild 1) sind mit dem Laborsystem zu erzeugen. Die Signale laufen periodisch weiter. Nutzen Sie den Channel 2 der TPU und die im Bild 1 genannten Portpins. Planen und schreiben Sie dafür ein C-Programm.

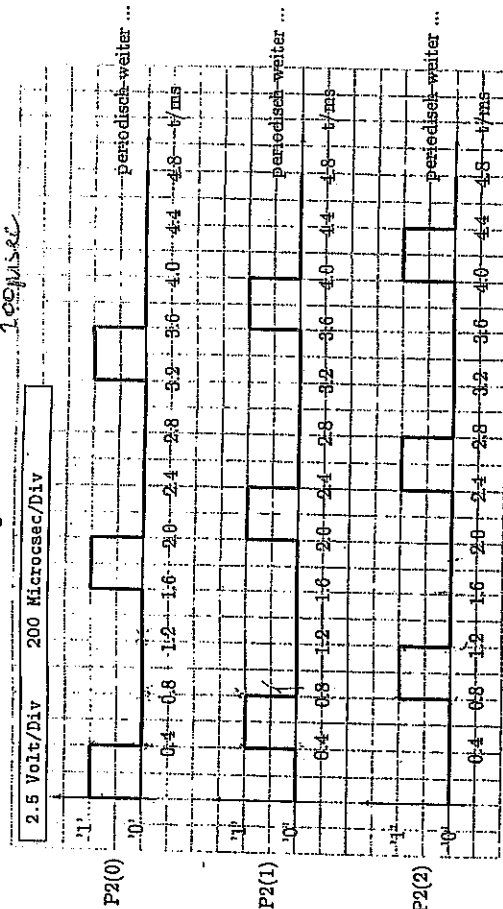


Bild 1: Mit dem Laborsystem zu erzeugende Signale

Bevor Sie programmieren, beantworten Sie bitte zunächst die Fragen:

Output compare (nicht!)
clear by TGRA, rising edge

a) Welchen Arbeitsmodus der TPU werden Sie nutzen?

b) Welchen Pre-Scaler (Vorteiler-Faktor) wählen Sie aus?

c) In welchem Register wird die benötigte Zeit eingestellt?

d) Welchen Zeiteintrag (Zahlenwert in 'Counter-Ticks') wählen Sie dafür?

$400 \cdot 10^{-6}$
 $54,2535 \cdot 10^5$

TPU.TGRA

7373-1

1 1 1 2

e) Schreiben Sie für die Erzeugung der Signale ein C-Programm mit genauen Kommentaren.

```
#include <stdio.h>
void main(void) {
    PADDR &= 0x07; /* P2(0) = P2(1), P2(2) sind Ausgangs */
    TPU_TCNTR2 = 0x0000; /* clear counter */
    TPU_TCR2 = 0x20; /* clearing TCR2, rising edge, clk / 100ns */
    TPU_TGR2A = 0x7373; /* 400ms */
    TPU_TSR2 &= 0x04; /* Clear TGFA */
    TPU_TSTR = 0x04; /* Timer Channel 2 Start */
    while (1) {
        P2DR = 0x01; /* P2(0) = 1 */
        while ((TPU_TSR2 & 0x01) == 0x00); /* wait until TFI
        P2DR = 0x02; /* P2(1) = 1 */
        TPU_TSR2 &= 0xFE; /* clear TGFA */
        while ((TPU_TSR2 & 0x01) == 0x00); /* waits until */
        P2DR = 0x04; /* P2(2) = 1 */
        TPU_TSR2 &= 0xFE; /* clear TGFA */
        while ((TPU_TSR2 & 0x01) == 0x00); /* waits until */
        P2DR = 0x00; /* alles auf Null */
        TPU_TSR2 &= 0xFE; /* clear TGFA */
        while ((TPU_TSR2 & 0x01) == 0x00); /* waits until */
    }
}
```

SS / WS	Semester	Fach	Dozent
---------	----------	------	--------

FSR - Klausurensammlung

3 Verschlüsseln und Entschlüsseln der seriellen Übertragung

Mit den Laborsystemen ist ein einfaches Ver- und Entschlüsselungssystem in C zu programmieren. Die Daten werden seriell mit dem SCI Channel 1 empfangen, und mit dem SCI Channel 2 weitergeleitet. Es werden dabei die Bitraten 9600 bit/s mit den Parametern 801 eingestellt.

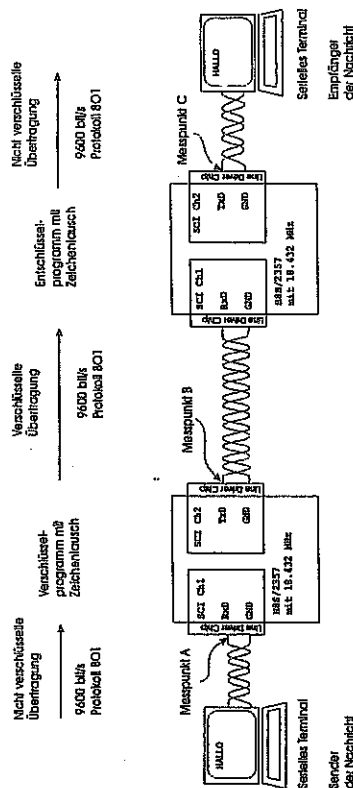


Bild 2: Übersicht zur verschlüsselten Übertragung

Für die Ver- und Entschlüsselung wird eine sehr einfache Codiervorschrift benutzt. Es sind nur Großbuchstaben ohne Umlaute erlaubt (ASCII Codes 'A' bis 'Z'), alle anderen Zeichen werden in das Zeichen '*' umgewandelt. In der Codiervorschrift wird jedes erlaubte Zeichen durch genau ein anderes Zeichen ersetzt. Dabei werden stets Tauschpaare der erlaubten Zeichen gebildet.

Für die Aufgabe sind folgende Tauschpaare vorgegeben:

- ('A' ↔ 'B')
- ('C' ↔ 'H')
- ('D' ↔ 'X')
- ('E' ↔ 'Z')

Die restlichen Tauschpaare sollen Sie frei wählen und zu einer individuellen Codiervorschrift ergänzen. Die Tauschpaarung erlaubt das Ver- und Entschlüsseln mit der gleichen Vorschrift.

¹ Ein Tauschpaar 'A' ↔ 'B' bedeutet: Ein 'A' wird durch ein 'B' ersetzt und ein 'B' wird durch ein 'A' ersetzt.
² Beispiel 'A' ↔ 'B': Ein 'A' wird durch ein 'B' beim Verschlüsseln ersetzt. Ebenso wird ein 'B' durch ein 'A' beim Entschlüsseln ersetzt. Das Entschlüsselgerät besitzt also das gleiche Programm wie das Verschlüsselgerät.

- a) Schreiben Sie die Funktion void main (void). Diese Funktion soll zunächst nur die Initialisierungen des SCI durchführen und dann zyklisch die Funktionen readscii, writesci2 und change benutzen. Diese drei Funktionen werden später programmiert. Nur die erlaubten Zeichen ('A' bis 'Z') sind zu tauschen. Für alle anderen ankommenden Zeichen hat die Funktion main() das Zeichen '*' zu senden.

```

#include <pp1.h> /* Include Headerfile v. Registernames */
char readscii(void); /* Prototype of a serial read function */
void writesci2(char); /* Prototype of a serial write function */
char change(char); /* Prototype of character change function */
char z;
void main(void) /* main function without parameters and return value */
{

```

```

    SCI2_SCR = 0x00; /* sperren senden v. Empfänger */

```

```

    SCI1_SCR = 0x00; /* ----- */

```

```

    SCI2_SMR = 0x60; /* 801 */

```

```

    SCI2_BRR = 59; /* 9600 bit/s */

```

```

    SCI2_SCR |= 0x20; /* Senden freigeben */

```

```

    SCI1_SMR = 0x60; /* 801 */

```

```

    SCI1_BRR = 59; /* 9600 bit/s */

```

```

    SCI2_SCR |= 0x10; /* Empfangen freigeben */

```

```

    while(1) {

```

```

        z = readscii();

```

```

        if ((z < 0x41) && (z > 0x5A))
            z = '*';

```

```

        z = change(z);

```

```

        writesci2(z);
    }

```

- b) Schreiben Sie eine Einlesefunktion char readscii(void), die ein Zeichen vom SCI Channel 1 seriell einliest und das Zeichen als Rückgabewert zurückgibt.

```

char readscii(void)
{

```

```

    unsigned char c;

```

```

    while ((SCI2_SSR & 0x40) == 0) /* wait for received char */

```

```

        c = SCI1_RDR; /* read and save to buffer */

```

```

    return c;
}

```

- c) Schreiben Sie eine Ausgabefunktion writesci2(char out), die ein an die Funktion übergebenes Zeichen mit dem SCI Channel 2 seriell ausgibt.

```

void writesci2(char out)
{

```

```

    while ((SCI2_SSR & 0x80) == 0) /* wait for TDRF = empty */
        SCI2_TDR = out; /* write

```

```

    SCI2_SSR &= ~0x80; /* clear TDRF and start transmission */
}

```

SS / WS	Semester	Fach	Dozent
FSR - Klausurensammlung 2			

d) Schreiben Sie die Funktion `char change(char in)`, die genau ein als Parameter übergebenes Zeichen mit einem anderen Zeichen (Rückgabewert) tauscht. Die Tabelle der Tauschpaare soll dort lokal, aber für die gesamte Programmlaufzeit initialisiert sein.

Lösungshinweis: Vermeiden Sie damit eine einfache Fallunterscheidung (if-else / switch-case oder dergleichen), suchen Sie eine einfache Lösung, welche die initialisierte Tabelle nutzt!

char change(char in) {

$\text{char out; char } \text{tos}[I] = \{A, B, C, H, E, Z\};$

```
switch(in) {
    case A: out = 'B'; break;
    case B: out = 'A'; break;
    case C: out = 'H'; break;
    case H: out = 'C'; break;
    case E: out = 'I'; break;
    case I: out = 'E'; break;
    case *: out = '*'; break;
    default: out = 'in+1';
}
```

```
} return out;
```

Ein Tauschpreis bestimmt.

Entschlossenung! Aufbruch!

SS / WS	Semester	Fach	Dozent
---------	----------	------	--------

FS2 - Klausurensummlung

$$\frac{1}{\text{bit density}} = 10^4 \text{ bits} \quad \text{density} = 1 \quad U = 0.1 \text{ M/Mean} \quad 1 \text{ Mbps} \times t$$

e) Mit dem Oszilloskop wird ein Signal an drei Messpunkten (Messpunkte A, B, C in Bild 2) aufgenommen. Das Zeichen 'C' wird vom Terminal gesendet. Zeichnen Sie die Signale. Ergänzen Sie die gesamte Beschriftung, kennzeichnen Sie sämtliche Bits mit logischen Werten und Abkürzungen.

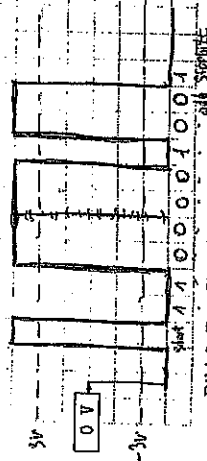
$$C = 0.100 \text{ eV}^{-1} \quad 1 \text{ Step } \frac{1}{2}$$


Bild 3: Das Signal für das Eingabezeichen 'C' vor der Verschlüsselung (Messpunkt A)

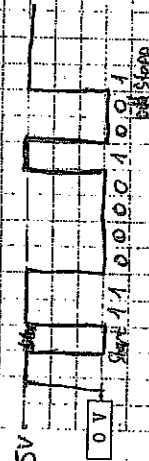


Bild 4: Das Signal für das Eingabezeichen 'C' nach der Verschlüsselung (Messpunkt B)

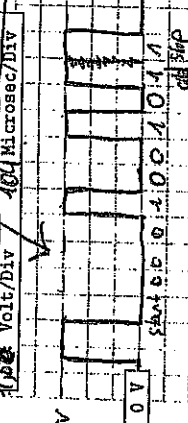


Bild 5 Das Signal für das Eingabezeichen 'C' nach der Entschlüsselung (Messpunkt C)

	...0	...1	...2	...3	...4	...5	...6	...7	...8	...9	...A	...B	...C	...D	...E
Gad...	NUL	SOR	STX	EXT	ROT	ENQ	AKN	BS	BT	EM	LF	VT	FF	CR	SO
Gai...	DLE	DCI	DC2	DC3	DC4	NAK	SYN	ETB	CAN	HT	SUB	ESC	FS	GE	US
Gac...	SP	1	#	#	\$	\$	\$	7	8	0	=	+	<	>	7
Gad...	0	1	2	3	4	5	6	7	8	9	1	2	3	4	5
Gae...	P	A	B	C	D	E	F	G	H	I	J	K	L	M	N
Gaf...	P	q	r	s	t	u	v	w	x	y	z	[\]	^
Gag...	p	q	r	s	t	u	v	w	x	y	z	[\]	^

Tabelle 1: ASCII Tabelle (niederwertige Bits in den Spalten, höherwertige Bits in den Zeilen)

Name, Vorname, Matr.Nr.: Konstantin Müller

30. Januar 2009 9

4 Fragenteil

a) Markieren Sie die richtige Aussage oder die richtigen Aussagen durch ankreuzen.

In einem von-Neumann-Rechner werden die Daten im RAM und das Programm im ROM gespeichert.	<input type="checkbox"/> zwingend notwendig	<input checked="" type="checkbox"/> möglich	<input type="checkbox"/> keinesfalls richtig	<input checked="" type="checkbox"/>
Ein Adresdecoder unterscheidet zwischen der Fetch- und Execute-Phase.	<input checked="" type="checkbox"/> richtig	<input type="checkbox"/> möglich	<input type="checkbox"/> falsch	<input checked="" type="checkbox"/>
Der interne ADG des H8S/2357 unterscheidet zwischen maximal wieviel Eingangswerten?	<input checked="" type="checkbox"/> 255	<input checked="" type="checkbox"/> 256	<input type="checkbox"/> 1024	<input checked="" type="checkbox"/>
Im Output-Compare-Mode der TPU werden die Werte welcher Komponenten verglichen?	<input checked="" type="checkbox"/> TIOCA u. TGFA	<input type="checkbox"/> TONT u. TGRx	<input type="checkbox"/> TSR u. TCR	<input checked="" type="checkbox"/>
Wenn ein übersetztes C-Programm ausgeführt wird, laufen eine Reihe Fetch- und Execute-Zyklen ab.	<input checked="" type="checkbox"/> richtig	<input type="checkbox"/> nur im Interrupthandler	<input type="checkbox"/> falsch	<input checked="" type="checkbox"/>
An dem SCI werden Daten in das Transmitter Shift Register (TSR) vom Programm unmittelbar eingetragen, wenn Daten gesendet werden sollen.	<input type="checkbox"/> ja	<input type="checkbox"/> Wenn die Bitrate nicht ausreicht	<input checked="" type="checkbox"/> nein	<input checked="" type="checkbox"/>
Ein Rechner mit 16 Bit breiten Datenbus kann mit wieviel Hauptspeicher maximal ausgestattet werden.	<input type="checkbox"/> 16 MByte	<input checked="" type="checkbox"/> keine Aussage möglich	<input type="checkbox"/> 4 GByte	<input checked="" type="checkbox"/>
Periphere-Register des H8S/2357 werden wie normale Hauptspeicherzellen vom Programm adressiert.	<input checked="" type="checkbox"/> ja	<input type="checkbox"/> wahlweise	<input checked="" type="checkbox"/> nein	<input checked="" type="checkbox"/>
Der Data Processor ist für die Steuerung der CPU-Funktion bei der Ausführung der Maschinenbefehle zuständig.	<input checked="" type="checkbox"/> nur in der Harvard-Architektur	<input checked="" type="checkbox"/> Neumann-Architektur	<input type="checkbox"/> minimals	<input checked="" type="checkbox"/>
Der Controller H8S/2357 hat welche Architektur (Harvard oder v. Neumann)?	<input checked="" type="checkbox"/> Harvard-Architektur	<input type="checkbox"/> von-Neumann-Architektur	<input type="checkbox"/> abhängig vom externen Speicher	<input checked="" type="checkbox"/>

b) Was kann einen Interrupt auslösen? Erklären Sie als kurze Stichpunkte.

... Interrupt löst einen Sprung im Programm aus.
 ... Nach dem Funktionsaufruf macht das Programm, das weiter, wo es gestoppt wurde.

c) Was wird bei einem Interrupt ausgeführt? Erklären Sie als kurze Stichpunkte.

... Eine Funktion die beim Eintreffen von Interrupt hinterlegt ist. Erst wenn diese beendet ist, kehrt man zum normalen Programm zurück.

SS / WS	Semester	Fach	Dozent
FSR - Klausurensammlung			

30.01.2009

Adressbereiche

Datum:

 Titel:
 ①

An einem Controller H8S/2357 sollen drei Speicherchips angeschlossen werden:

1. Das erste Speicherchip ist ein ROM-Chip mit 2 Megabyte Kapazität. Der Adressbereich des ROM-Chip soll an den untersten Adressen des Controllers ab 0x000000 liegen.
2. Der 2. Speicherchip ist ein EEPROM-Chip mit 2 Megabyte Kapazität. Der Adressbereich des EEPROM-Chip soll ab der Controller-Adresse 0x800000 beginnen.
3. Der 3. Speicherchip ist ein RAM-Chip mit 2 Megabyte Kapazität. Der Adressbereich dieses RAM-Chip umfasst die obersten Adressen des Controllers und endet mit der Adresse 0xFFFFF.

a) Welche Adressbereiche bleiben frei für spätere Erweiterungen? (Nennen Sie die freie Kapazität in MB, die Anfangs u. Endadresse (hexadezimal)).

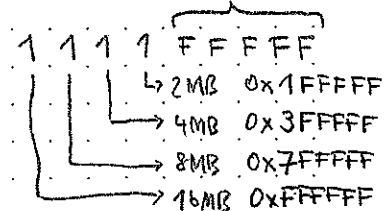
Kapazität	Anfangsadresse	Endadresse
6 MB	0x200000	0x7FFFFF
4 MB	0xA00000	0xDFFFFF

$$2 \text{ MB} = 2^{10} \cdot 2^{10} \cdot 2^1 = 2^{21} \rightarrow A_{20}$$

 A23 A22 A21 A20
 0 0 0 1

 $9 = 1001$
 $A = 10 = 1010$
 $B = 11 = 1011$
 $C = 12 = 1100$
 $D = 13 = 1101$
 $E = 14 = 1110$
 $F = 15 = 1111$

A23 A22 A21 A20 Rest


 b) Geben Sie die Gleichungen des Chip Select (\overline{CS}) vom ROM, EEPROM, u. RAM an, wenn diese aus den Adressleitungen des Controllers erzeugt werden.

ROM-Chip: $CS_{ROM} = A_{23} \wedge A_{22} \wedge A_{21}$

0x000000 ... 0x1FFFFF

$$\overline{CS}_{ROM} = A_{23} \vee A_{22} \vee A_{21} \quad \left[\overline{CS}_{ROM} = \overline{A_{23} \wedge A_{22} \wedge A_{21}} \right]$$

EEPROM-Chip: $CS_{EEPROM} = A_{23} \wedge A_{22} \wedge A_{21}$

0x800000 ... 0x9FFFFF

$$\overline{CS}_{EEPROM} = A_{23} \wedge A_{22} \wedge A_{21}$$

[1000]0

[1001]1

$$\overline{CS}_{EEPROM} = A_{23} \vee A_{22} \vee A_{21}$$

RAM-Chip: $CS_{RAM} = A_{23} \wedge A_{22} \wedge A_{21}$

0xE00000

0xFFFFF

$$\overline{CS}_{RAM} = A_{23} \wedge A_{22} \wedge A_{21}$$

[1110]0

[1111]1

$$\overline{CS}_{RAM} = A_{23} \vee A_{22} \vee A_{21}$$



Wusstest du,

...dass Termiten sich doppelt so schnell durch einen Baum fressen, wenn sie Rockmusik hören?

StudiBlock Partner:

otto group

30.01.2009

Periodische Signalfolgen mit der TPU erzeugen

StudiBlock

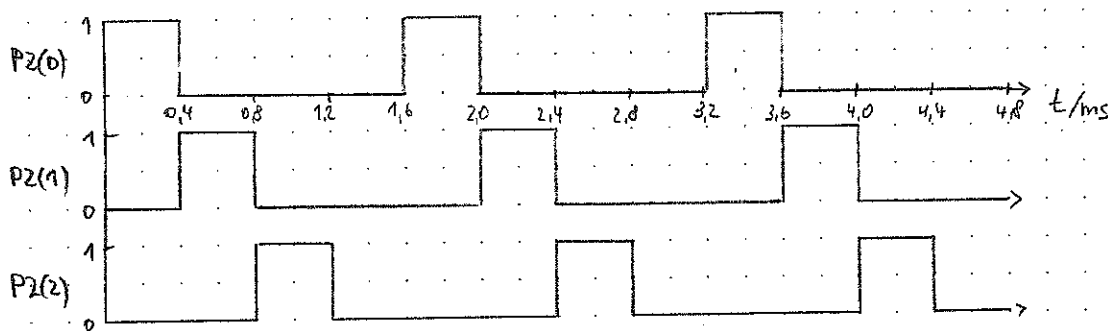
Stoff für die Vorlesung

periodisch

Titel:

Datum:

- ② Die folgenden Signale (Bild 1) sind mit dem Laborsystem zu erzeugen. Die Signale laufen periodisch weiter. Nutzen Sie den Channel 2 der TPU und die in Bild 1 genannten Portpin-Pläne u. schreiben Sie dafür ein C-Programm.



$$200 \mu\text{s/div} = X$$

$$2.5 \text{ V/div} = Y$$

Bevor Sie programmieren, beantworten Sie zunächst die Fragen:

- a) Welchen Arbeitsmode der TPU werden Sie nutzen?

output compare, clear by TGRA; rising edge

- b) Welchen Pre-Scaler (Vorfaktor-Faktor) wählen Sie aus?

1

um Ticks maximal zu bekommen aber unter 65535 zu bleiben, kann Prescaler variiert werden! (1, 4, 16, 64)

- c) In welchem Register wird die benötigte Zeit eingestellt?

$$\text{Ticks} = \frac{1}{P} \cdot \text{CLK} < 65535!$$

TPU-TGR2A

- d) Welchen Zeiteintrag (Zahlenwert in Counter-Ticks) wählen Sie dafür?

$$\frac{0.4 \text{ ms}}{54.2535 \text{ ns}} = 7373 - 1$$

$$7373 - 1$$

- e) Schreiben Sie für die Erzeugung der Signale ein C-Programm mit genauen Kommentaren!

```
#include <msp1.h> // Registernamen aus Header file liefern
void main(void) { // main function
    P2DDR &= 0x07; // PZ(0), PZ(1), PZ(2) sind Ausgang

    TPU-TCNT2 = 0x0000; // clear counter

    TPU-TCR2 = 0x20; // clear by TGRA; rising edge, clk/1

    TPU-TGR2A = (7373-1); // T=400 μs →  $\frac{400 \cdot 10^{-6}}{54.25 \cdot 10^{-9}} = 7373$ 

    TPU-TSR2 &= 0xFE; // clear TGRA

    TPU-TSTR = 0x04; // start Timer Channel 2
```

SS, WS	Semester	Fach	Doxent
--------	----------	------	--------

FSR - Klausurensammlung 7

StudiBlock Partner:

otto group



Wusstest du,

...dass die Tragzeit einer Rhinoceros-Kuh unglaubliche 560 Tage dauert?



weiter bei 2)

```

while (1) { // Endloschleife

    P2DR = 0x01; // P2(0) auf '1' setzen
    TPU-TSR2 &= 0xFE; // Clear TGFA
    while ((TPU-TSR2 & 0x01) == 0x00); // Polling auf TGFA (wait 400µs)

    P2DR = 0x02; // P2(1) auf '1' setzen
    TPU-TSR2 &= 0xFE; // Clear TGFA
    while ((TPU-TSR2 & 0x01) == 0x00); // Polling auf TGFA (wait 400µs)

    P2DR = 0x04; // P2(2) auf '1' setzen
    TPU-TSR2 &= 0xFE; // Clear TGFA
    while ((TPU-TSR2 & 0x01) == 0x00); // Polling auf TGFA (wait 600µs)

    P2DR = 0x00; // P2 auf Null setzen
    TPU-TSR2 &= 0xFE; // Clear TGFA
    while ((TPU-TSR2 & 0x01) == 0x00); // Polling auf TGFA (wait 400µs)
} // Ende while-Schleife
} // Ende main function
  
```

SS / VS	Semester	Fach	Dozent
---------	----------	------	--------

FSR - Klausurensammlung 8



Wusstest du,

...das Moskitos 47 Zähne haben?

StudiBlock Partner:

otto group

30.01.2009

Verschlüsseln u. Entschlüsseln des seriellen

Datum:

Thema:
3

Übertragung

- ② Schreiben Sie die Funktion `void main(void)`. Diese Funktion soll zunächst nur die Initialisierung der SCI durchführen u. dann zyklisch die Funktionen `readsci1`, `writesci2` und `change` benutzen. Diese drei Funktionen werden später programmiert. Nur die erlaubten Zeichen ('A' bis 'Z') sind zu tauschen. Für alle anderen ankommenden Zeichen hat die Funktion `main()` das Zeichen '*' zu senden.

```
#include <uip.h> /* include header file uip. Register names */
char readsci1(void); /* prototype of a serial read function */
void writesci2(char); /* write function */
char change(char); /* character change function */
char z;
void main(void) /* main function without parameters and return value */
{
    SCI2_SCR = 0x00; /* sperren senden u. empfangen */
    SCI1_SCR = 0x00; /* sperren senden u. empfangen */
    SCI2_SMR = 0x30; /* 801 */
    SCI2_BRR = 59; /* 9600 bit/s */
    SCI2_SCR |= 0x20; /* senden freigeben */
    SCI1_SMR = 0x30; /* 801 */
    SCI1_BRR = 59; /* 9600 bit/s */
    SCI1_SCR |= 0x10; /* Empfang freigeben */
    while(1) { /* Endlosschleife */
        z = readsci1(); /* Einflesefunktion aufrufen u. Wert auf z übertragen */
        if((z < 0x41) && (z > 0x5A)) /* Abfrage, ob Großbuchstaben oder anderes Zeichen */
            z = "*"; /* wenn kein Großbuchstabe -> mit * auffüllen */
        z = change(z); /* change Funktion aufrufen u. Wert auf z übertragen */
        writesci2(z); /* Ausgabe funktion aufrufen */
    } /* Ende while-Schleife */
} /* Ende main function */
```

SS / WS	Semester	Fach	Dozent
---------	----------	------	--------

FSR - Klausurensammlung 9



Wusstest du,

...dass das längste deutsche Wort, in dem kein Buchstabe zwei Mal vorkommt

„Heizölrückstoßabdämpfung“ ist?

StudiBlock Partner:

otto group

Teil:
 3

- b) Schreiben Sie eine Eingabefunktion `char readsci1(void)`, die ein Zeichen vom SCI Channel 1 seriell einliest und das Zeichen als Rückgabewert zurückgibt.

```

char readsci1(void)
{
    unsigned char c;
    while ((SCI1-SSR & 0x40) == 0) /* wait for received char */
        ;
    c = SCI1-RDR; /* read and save to buffer */
    return c; /* Rückgabewert
  
```

- c) Schreiben Sie eine Ausgabe funktion `writesci2(char out)`, die ein an die Funktion übergebenes Zeichen mit dem SCI Channel 2 seriell ausgibt.

```

void writesci2(char out)
{
    while ((SCI2-SSR & 0x80) == 0) /* wait for TDRE = empty */
        ;
    SCI2-TDR = out; /* write */
    SCI2-SSR &= ~0x80 /* clear TDRE and start Transmission */
  
```

SS / WS	Semester	Fach	Dozent
---------	----------	------	--------

FSR - Klausurenansammlung 10



Wusstest du,

...dass etwa 80% aller Tiere auf der Erde sechs Beine haben?

StudBlock Partner:

otto group

- d) Schreiben Sie die Funktion `char change(char in)`, die genau ein als Parameter übergebenen Zeichen mit einem anderen Zeichen (Rückgabewert) tauscht. Die Tabelle der Tauschpaare soll dort lokal, aber für die gesamte Programmlaufzeit initialisiert sein.

Lösungshinweis: Vermeiden Sie damit eine einfache Fallunterscheidung (if-else / switch-case oder dergleichen), suchen Sie eine einfache Lösung, welche die initialisierte Tabelle nutzt.

```
char change(char in)
{
```

?

SS / WS	Semester	Früh	Spät
FSR - Klausurensammlung			



Wusstest du,

...dass die Seidenspringerraupe tatsächlich 11 Gehirne hat?

StudiBlock Partner:

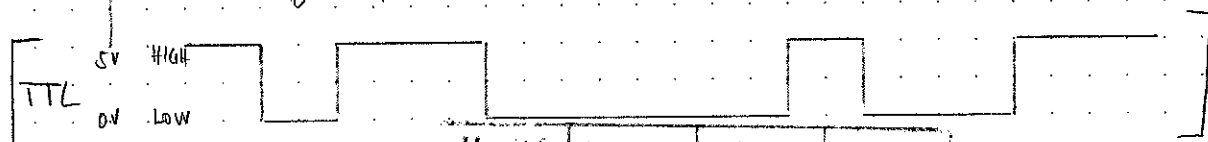
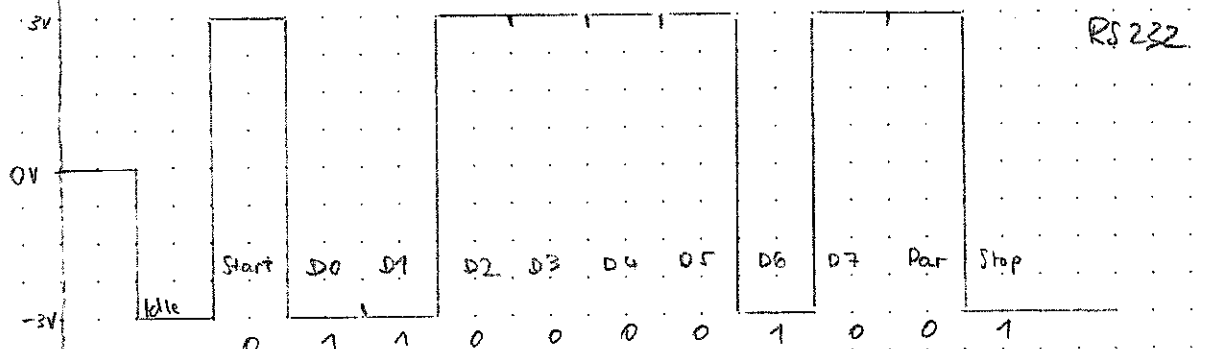
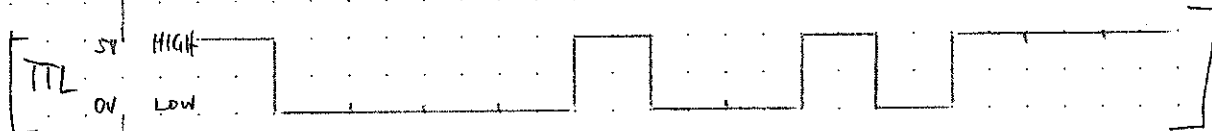
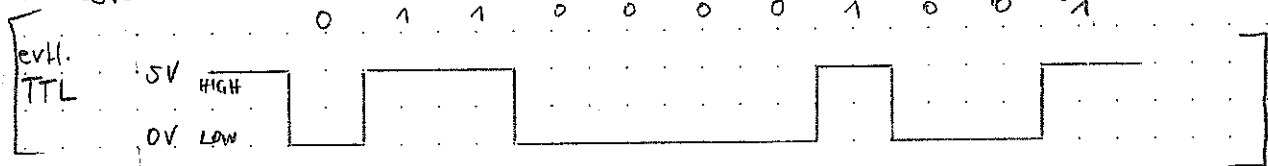
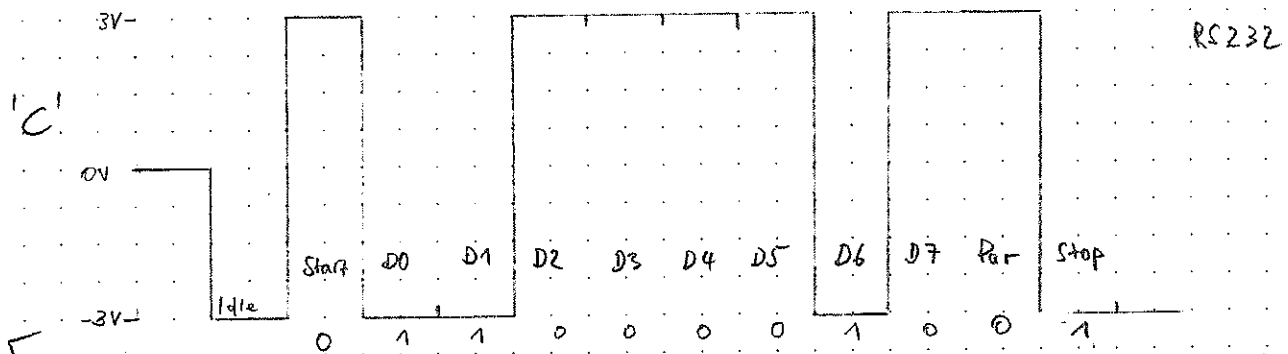
otto group

③
 Titel:

- © Mit einem Oszilloskop wird ein Signal an drei Messpunkten (Messpunkt A, B, C in Bild 2) aufgenommen. Das Zeichen 'C' wird vom Terminal geschickt. Zeichnen Sie die Signale. Ergänzen Sie die gesamte Beschriftung, kennzeichnen Sie sämtliche Bits mit logischen Werten u. Abkürzungen.

C = 0100 0011 Odd 1 Stopbit 0x43

H = 0100 1000 Odd 1 Stopbit 0x48



SS / WS	Semester	Fach	Version
FSR - Klausurensammlung			



Wusstest du,

...dass sich die Römer mit Vogelkot die Haare blond färbten?

StudiBlock Partner:

otto group

4er

a) Markieren Sie die richtige Aussage oder die richtigen Aussagen durch ankreuzen.

In einem von-Neumann-Rechner werden die Daten im RAM u. den Programmen im ROM gespeichert.	<input type="checkbox"/> zwingend notwendig	<input checked="" type="checkbox"/> möglich	<input type="checkbox"/> keinesfalls richtig	✓
Ein Adressdecoder unterscheidet zwischen der Fetch u. Execute Phase.	<input type="checkbox"/> richtig	<input type="checkbox"/> möglich	<input checked="" type="checkbox"/> falsch	?
Der interne ADC des H8S/2357 unterscheidet zwischen maximal wievielen Eingangswerten?	<input checked="" type="checkbox"/> 255	<input type="checkbox"/> 256	<input checked="" type="checkbox"/> 1024	?
Im Output-Compare-Mode des TPU werden die Werte welcher Komponenten verglichen?	<input type="checkbox"/> TI0CA u. TQFA	<input checked="" type="checkbox"/> TCNT u. TGRX	<input type="checkbox"/> TSR u. TCR	✓ +
Wenn ein übersetztes C-Programm ausgeführt wird, laufen eine Reihe von Fetch- u. Execute-Zyklen ab.	<input checked="" type="checkbox"/> richtig	<input type="checkbox"/> nur im Interpreter	<input type="checkbox"/> falsch	✓
An dem Schieben Daten in das Transmitter Shift Register (TSR) vom Programm unmittelbar eingetragten, wenn Daten gerundet werden sollen.	<input type="checkbox"/> ja	<input type="checkbox"/> Wenn die Bitrate nicht anwächst	<input checked="" type="checkbox"/> nein	✓
Ein Rechner mit 16 Bit breiter Datenbus kann mit wieviel Hauptspeicher maximal ausgestattet werden.	<input type="checkbox"/> 16 MByte	<input checked="" type="checkbox"/> keine Aussage möglich	<input type="checkbox"/> 4 GByte	✓
Peripherie-Register des H8S/2357 werden wie normale Hauptspeicherseiten vom Programm adressiert.	<input checked="" type="checkbox"/> ja	<input checked="" type="checkbox"/> teilweise	<input type="checkbox"/> nein	
Der Data Prozessor ist für die Steuerung der CPU-Funktion bei der Ausführung der Maschinenbefehle zuständig.	<input type="checkbox"/> nur in d. Harvard-Architektur	<input type="checkbox"/> nur in der von Neumann-Architektur	<input checked="" type="checkbox"/> niemals	?
Der Controller H8S/2357 hat welche Architektur (Harvard od. v. Neumann)?	<input type="checkbox"/> Harvard-Architektur	<input checked="" type="checkbox"/> von-Neumann-Architektur	<input type="checkbox"/> abhängig vom externen Speicher	✓ +

b) Was kann einen Interrupt auslösen? Erklären Sie als kurze Stichpunkte.

- ein Interrupt ist eine Exception, die von einem externen Ereignis (externer Interrupt) oder durch eine chipinterne Peripheriefunktion (interner Interrupt) angefordert wird.
- Bsp.: Reset, ext. Interrupt Anforderung, innerhalb eines Prozesses (Trace Exception)

c) Was wird bei einem Interrupt ausgeführt? Erklären Sie als kurze Stichpunkte.

- Wenn ein Interrupt (Exception) auftritt, wird ein ^(Interrupt) Exception-Handler ausgeführt.
- Dieser Exception Handler ist ein spezielles Programm, das für die Reaktion auf das Ereignis, das die Exception (Interrupt) verursacht hat, verantwortlich ist.

SS	VS	Lehrstuhl	Fach	Dozent
FSR - Klausurensammlung 13				



Wusstest du,

...dass jeder Elefant pro Tag durchschnittlich zwei Stunden schläft?

StudiBlock Partner:

otto group