



Technische
Universität
Braunschweig

Institut für Betriebssysteme
und Rechnerverbund



Programmierung verteilter eingebetteter Systeme

Teamprojekt – Einführung und Grundlagen Teil 1

Stephan Rottmann, Ulf Kulau, Felix Büsching
Summer Term 2014

- **Organisatorisches**
- Grundlagen 1
- Praktikumshardware
- Aufgabe 1

Teilnehmerinnen und Teilnehmer

13 Anmeldungen – 5 Gruppen

- Gruppe 1: Sascha, Tobias
- Gruppe 2: Heiner, Adrian
- Gruppe 3: Fabian, Dennis, David
- Gruppe 4: Arne, André, Lars
- Gruppe 5: Stefanie, Marian, Quentin

Raumnutzung und Belegung: Raum 148 (hier)

Praktikumsraum

- 3 Arbeitsplätze für je 3 Personen
 - Nicht essen -> macht eklige Tastaturen
 - Möglichst nichts verschütten
- **Dieses Praktikum**
 - Hiwis anwesend
 - Gruppen 1, 2, 4: Montags, 13:15 – 16:15
 - Gruppen 3: Mittwochs, 11:15 – 14:15
 - Gruppen 5: Donnerstags, 13:15 – 16:15
- **Anderes Praktikum**
- **Freie Zeit bei Bedarf**
 - Ggf. im Miclab fragen, ob man euch in den Raum lässt

Uhrzeit	Mo	Di	Mi	Do	Fr
8:00-9:30					
9:45-11:15					
11:30-13:00			3		
13:15-14:45	1,2,4		3	5	
15:00-16:30	1,2,4			5	
16:45-18:15					
18:30-20:00					

Raumnutzung Raum 147 (→) & Werkstatt (Keller)

Mikroprozessorlabor

- 2 Lötarbeitsplätze
- Umfangreiche Sammlung an Bauteilen für eigene Ideen
- Menschen mit Erfahrung

Werkstatt

- Platinen belichten, entwickeln, ätzen und bohren
- Mechanische Bearbeitung von allem möglichen

Betreuung

- Jost Stolze, Montag ab 13:15
- Dennis Reimers, Mittwoch ab 11:30
- Karsten Hinz, Donnerstag ab 13:15

- Ulf Kulau, IZ Raum 111
- Stephan Rottmann, IZ Raum 118
- Felix Büsching, IZ Raum 132

Accounts und Zugänge

Rechner-Login

- Gruppenaccounts (lokal an den Rechnern, daher immer den gleichen PC nutzen)

Benutzer-Accounts

- <http://www.ibr.cs.tu-bs.de/passwd/rz.html>
- Y-Nummer registrieren

Wiki

- <https://trac.ibr.cs.tu-bs.de/course-cm-ss14-esys/wiki/>
- Dokumentation, Kollaboration, Diskussion
- y-Nummer bitte **jetzt** eintragen

SVN-Zugang

- <https://svn.ibr.cs.tu-bs.de/course-cm-ss14-esys/>
- Alles ist im SVN zu speichern, hierfür y-Account nutzen!

Ablauf des Praktikums

Aufgabenphase

- Start: sofort
- Aufgaben werden in den Teams bearbeitet
- Hiwis geben gerne Hilfestellungen
- 2 Kolloquien je Gruppe über die gelernten Dinge
- Ende: Nach Bearbeitung der Aufgaben
 - ≥ 1 Aufgabe pro Termin

Projektphase

- Start: Nach Aufgabenphase
- Eigenes Projekt im Team oder gemeinsam mit mehreren Teams
- Ende: Ende des Semesters (nicht Ende der Vorlesungszeit)

- Organisatorisches
- Grundlagen 1
- Praktikumshardware
- Aufgabe 1

Eingebettete Computer sind allgegenwärtig



Verkehr

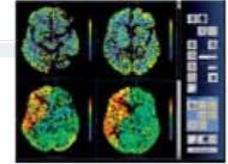
- Haus- und Gebäudetechnik
- Industrieelektronik
- Sicherheitstechnik
- Raumfahrttechnik
- ...



Unterhaltungselektronik



Kommunikation



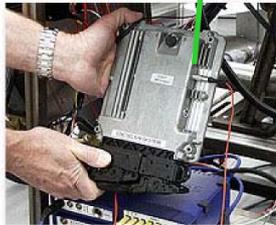
Medizintechnik



Finanzwesen

Quellen: Siemens, Toyota, Sony, u.a.

Eingebettete Computer mit verschiedenen Eigenschaften



robuste Computer
(Motorsteuergerät)



stromsparende
Computer
(Mobilkommunikation)



fehlertolerante Computer
(Raumfahrt)



leistungsfähige Computer
(z.B. Medizintechnik,
Graphik)

zugriffsgeschützte
Computer
(SmartCard)



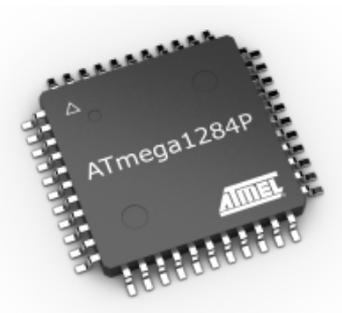
ganz kleine
Computer
(RFID)



Microcontroller

Microcontroller = Microprocessor + Peripherals

- Memory
 - Main-, program- and data-memory
 - SD-RAM, Flash, EEPROM
- Bus-controller
 - USART, UART, I²C, SPI, DMA, USB, CAN, Ethernet
- GPIO
- ADC / Comparators, DAC
- Timer



Reale Welt vs. Rechner-Welt

Realität: Analog

- Unendlich viele Farben
- Unendlich große Zahlen
- Irrationale und nicht-deterministische Entscheidungen

Rechner-Welt: Digital / Binär

- Begrenzte Zahlenräume
- I.d.R. Deterministisches Verhalten (gewünscht)
- Keine 100%ig genaue Abbildung der Realität möglich
 - Immer nur Annäherungen

Rechner: Darstellung von Zahlen durch Spannungen

Letztendlich geht es immer um das Rechnen mit Zahlen!

- Rechner arbeiten im Binärsystem: Es existieren nur „0“ und „1“
- Physikalisch passiert das durch unterschiedliche Spannungspegel
 - Bspw: 0V == „0“, 5V == „1“

Logikpegel nicht absolut

Technologie	Low (V_{IL})	High (V_{IH})
<u>TTL</u> 5V	$\leq 0,8$	$\geq 2,0$
<u>CMOS</u> 5V	$\leq 1,5$	$\geq 3,5$
LVTTL 3,3V	$\leq 0,8$	$\geq 2,0$
CMOS 2,5V	$\leq 0,7$	$\geq 1,7$
CMOS 1,8V	$\leq 0,7$	$\geq 1,17$
RS-232	-15 bis -3	+3 bis + 15

Zahlensysteme

Wir rechnen (meist) im 10er-System

- 10 Symbole: 0 bis 9
- Größere Zahlen brauchen mehr Stellen:
 - $423_{10} = 4 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0$

Computer rechnen immer im 2er-System

- 2 Symbole: 0 und 1
- Stellen haben Wertigkeit 1, 2, 4, 8, ... ($2^0, 2^1, 2^2$)
- $5_{10} = 1 \cdot 4 + 1 \cdot 1 = 1 \cdot 2^2 + 1 \cdot 2^0 = 101_2$

Weitere Zahlensysteme

Bei der Programmierung findet man häufig die Hexadezimaldarstellung, Basis 16

- 0, 1, ..., 9, A, B, ..., F
- Zahlen lassen sich mit Basis 2 konvertieren:

0 x 3 2 A

0 b 0 0 1 1 0 0 1 0 1 0 1 0

Gelegentlich wird das oktale System genutzt

- Basis 8

Zahlensysteme in C

Basis wird mittels Präfix angegeben

Basis	Präfix	Beispiel für Zahl 23_{10}
2	0b	0b10111
8	0	027
10	(n/a)	23
16	0x	0x11

Achtung: 0123 in C entspricht 83_{10}

Kombination möglich, aber nicht immer sinnvoll

▪ `int c = 0x22 + 011 + 0b1011`

Bitoperationen

(N)AND

(N)OR

XOR

NOT

SHIFT

Das N steht für NOT und dreht Ergebnis um

AND

Ver-UND-ung

A	B	A AND B	A NAND B
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

A	1	0	0	0	1	1	0	1
AND								
B	1	1	0	1	1	0	1	1
=	1	0	0	0	1	0	0	1

OR

Ver-ODER-ung

A	B	A OR B	A NOR B
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

A	1	0	0	0	1	1	0	1
OR								
B	1	1	0	1	1	0	1	1
=	1	1	0	1	1	1	1	1

XOR

Exklusives ODER

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

A	1	0	0	0	1	1	0	1
XOR								
B	1	1	0	1	1	0	1	1
=	0	1	0	1	0	1	1	0

NOT

Negierung, unär

A	<i>NOT A</i>
0	1
1	0

SHIFT

Verschieben von Werten

```
int a = 7;    //0b00000111
int b;

b = a << 2;   //0b00011100
```

Anwendung

Alle Operationen in C **bitweise** oder **logisch** verwendbar

Logisch: Verwendung bei Abfragen (true/false)

Typ	bitweise	logisch
AND	<code>a & b</code>	<code>a && b</code>
OR	<code>a b</code>	<code>a b</code>
XOR	<code>a ^ b</code>	(n/a)
NOT	<code>~</code>	<code>!</code>
SHIFT	<code><< , >></code>	(n/a)

Einsatzgebiete

AND: Testen, ob Bit gesetzt

OR: Ein Bit setzen

XOR: Ein Bit wechseln

NOT: Negierung

SHIFT: Bits verschieben

Codebeispiel

```
int bit3 = 1 << 3; // 0b00001000

if(PIND & bit3)
{
    //Bit 3 an PIND ist gesetzt
    printf("Bit %d ist gesetzt\n", 3);
}

PORTB |= bit3; // Bit 3 in PORTB setzen
PORTB ^= (1<<2); // Bit 2 in PORTB wechseln

PORTC &= ~(1<<3); //Bit 3 in PORTC löschen
```

(Noch) Advanced, aber bald hilfreich

Datenblatt: Register und enthaltene Bits

Diese sind als Makros in C verfügbar

Port D Data Register – PORTD

Bit	7	6	5	4	3	2	1	0	
	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

Port D Data Direction Register – DDRD

Bit	7	6	5	4	3	2	1	0	
	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

```
//Setup Data Direction Register
```

```
DDRD = (1<<DDD7) | (1<<DDD5) | (1<<DDD3);
```

```
//Turn on PORTD7, Turn off PORTD1
```

```
PORTD |= (1<<PORTD7);
```

```
PORTD &= ~(1<<PORTD1);
```

Digitalelektronik

Digitale Elektronik kennt zwei Zustände: 1 und 0
Zustände entsprechen i.d.R. auf Spannungen

- z.B. 1 → 3,3V / 0 → 0V

Ein **Bit** kann Wert 0 oder 1 annehmen

Ein **Byte** besteht aus 8 **Bit**

Ein **8 Bit-Mikrocontroller** kann **8 Bit** gleichzeitig verarbeiten

- Zahlen mit mehr als 8 Bit werden in mehreren Schritten verarbeitet

Schnittstelle zwischen Soft- und Hardware

Mikrocontroller können über I/O-Pins mit der Welt interagieren:

Digitale Werte ausgeben oder einlesen

Ein Port hat (meist) 8 Bit

Ein Pin entspricht einem Bit

Pins können (meist) als Ein- oder Ausgang konfiguriert werden

Interaktion über Ports und deren Pins

Ein Schalter kann an einen als Eingang geschalteten Pin angeschlossen und von der Software ausgelesen werden

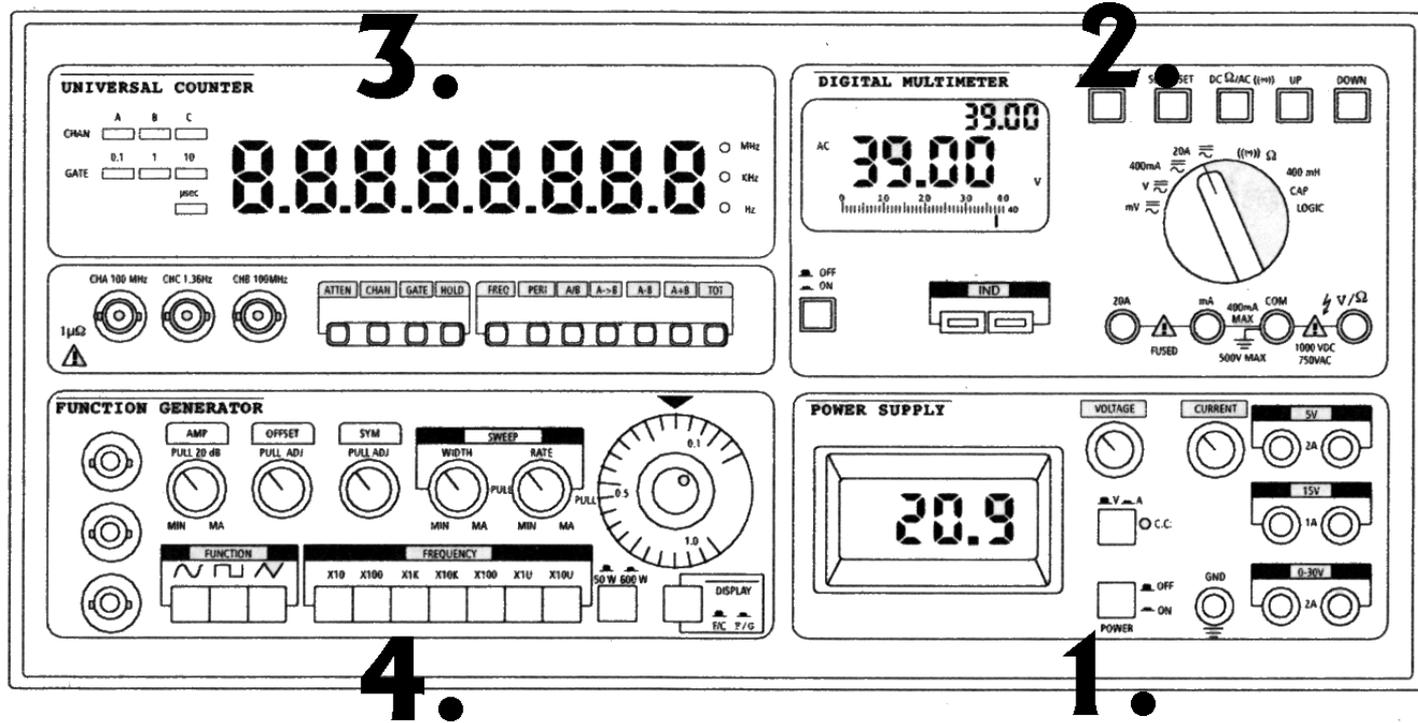
LEDs oder Transistoren für größere Leistungen können ein- und ausgeschaltet werden

- Bit für Pin gesetzt: Spannung liegt an

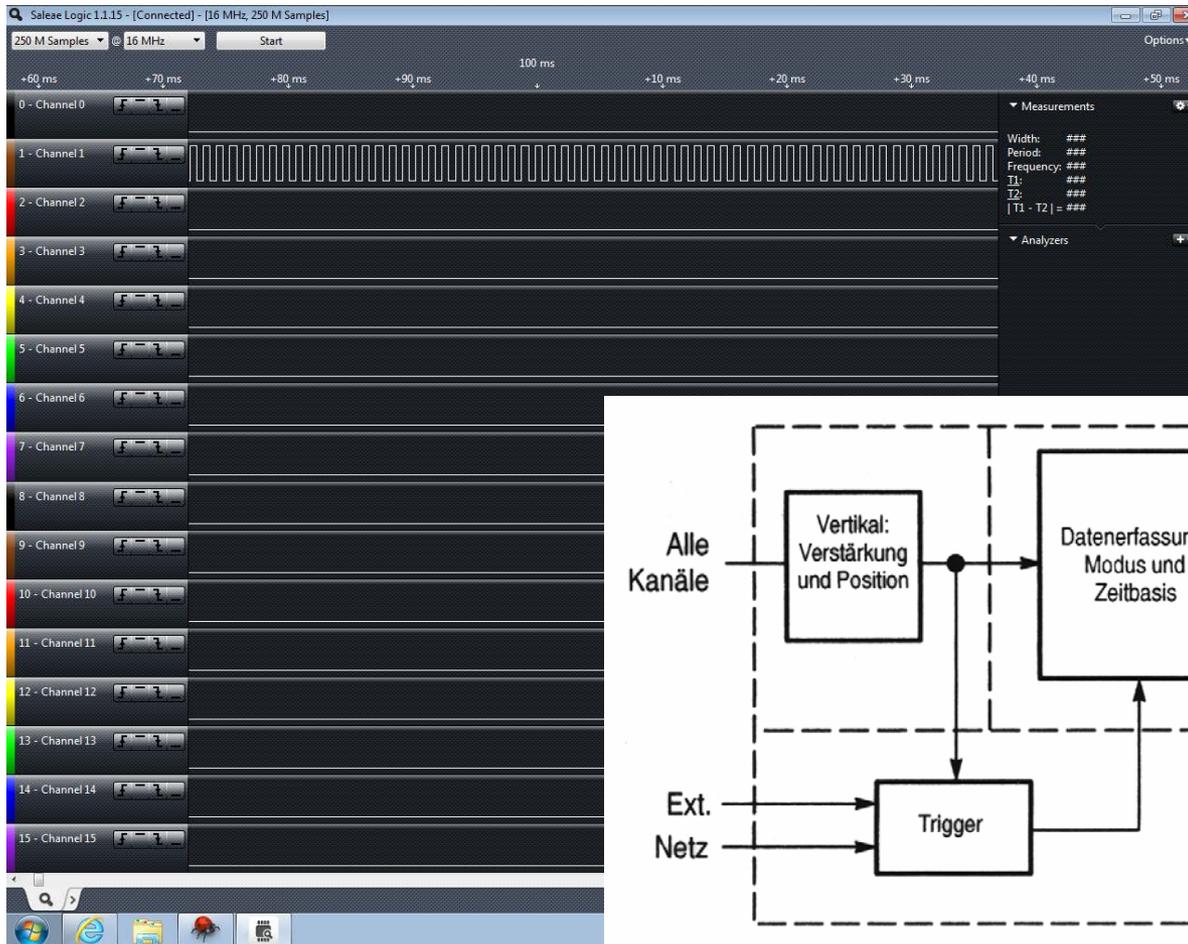
Digitale Bausteine können angeschlossen werden

- Organisatorisches
- Grundlagen 1
- **Praktikumshardware**
- Aufgabe 1

Mess-Station

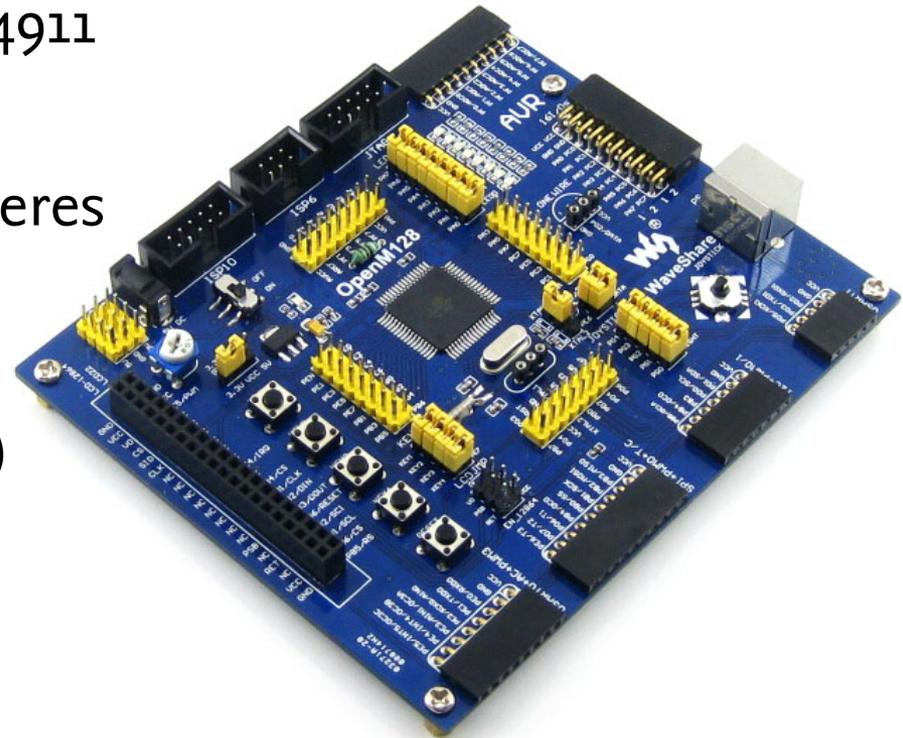


Logikanalysator & digitales Speicheroszilloskop



Entwicklungsboard

- Temperatursensor für I2C-Bus DS1621
- I2C-EEPROM AT24C0x
- SPI-Digital-Analogwandler MCP4911
- LC-Display
- Funkmodule: Bluetooth und anderes auf 2,4GHz
- weitere Speicher, Displays, Eingabemodule (Tastenfelder, ...) ... und vieles mehr



- Organisatorisches
- Grundlagen 1
- Praktikumshardware
- **Aufgabe 1**