

Institutsleitung

Prof. Dr.-Ing. Dr. h. c. J. Becker

Prof. Dr. rer. nat. W. Stork

Prof. Dr.-Ing. E. Sax

Prof. Dr.-Ing. E. Sax

M. Sc. Jijing Yan

Institut für Technik der Informationsverarbeitung (ITIV)



**Einführung in das Praktikum
Informationstechnik**

Inhalt



1

- **Organisatorisches**

2

- **Ablauf des Praktikums**

3

- **Bewertung**

4

- **Einführung des Auftraggebers**

5

- **Einführung in die Aufgabenstellung**

Vorlesung

- Erlernen der Theorie und des allgemeinen Verständnisses zum Programmieren und für Algorithmen und Datenstrukturen, Einführung in die Informatik

Übung

- Erlernen der Programmiersprache C++ und praktisches Anwenden der Inhalte aus der Vorlesung



Praktikum

- Praktisches Anwenden der Kenntnisse anhand eines größeren Projekts und Vertiefung der Programmierkenntnisse

Ziele des Praktikums

- **Praktische Anwendung der Vorlesungsinhalte**
 - Passende Algorithmen und Datenstrukturen entsprechend einer bestimmten Aufgabenstellung anwenden
- **Praktische Anwendung der Übungsinhalte**
 - Umsetzung in hardwarenahen C++ Code
 - Programm strukturiert aufbauen und die Befehle zur Umsetzung der Anforderungen bestimmten Funktionen zuordnen
- **Praktische Erfahrung im Projekt (Teamarbeit und Softwareentwicklung)**
 - Zerlegen von komplexen Problemen, dargeboten in natürlicher Sprache (Spezifikation), in einfache und übersichtliche Module, sowie Ausdrücken dieser Module mit Hilfe einer Programmiersprache (C++)
 - Erzeugen von hardwarenahem Code unter Einhaltung vorgegebener Qualitätskriterien (Programmierrichtlinien)
 - Einhalten eines vorgegebenen Zeitplans

IT-Veranstaltungsplan SS2018

KW	SW	Datum Mittwoch	9:45 – 11:15 Uhr (Benz)	Datum Donnerstag	14:00 – 15:30 Uhr (Neue Chemie)	Praktikum
16.	1.	18. Apr	VL	19. Apr		
17.	2.	25. Apr	VL	26. Apr		
18.	3.	02. Mai	VL	03. Mai	ÜB	
19.	4.	09. Mai	VL	10. Mai	fei	P (EV-I)
20.	5.	16. Mai	VL	17. Mai		P (EV-II)
21.	6.	23. Mai	frei	24. Mai	frei	
22.	7.	30. Mai		31. Mai	frei	P
23.	8.	06. Jun	VL	07. Jun	ÜB	P
24.	9.	13. Jun	VL	14. Jun	ÜB	P
25.	10.	20. Jun	VL	21. Jun	VL	P
26.	11.	27. Jun	VL	28. Jun	ÜB	P
27.	12.	04. Jul	VL	05. Jul	ÜB	P
28.	13.	11. Jul	VL	12. Jul	ÜB	P (PF)
29.	14.	18. Jul	VL	19. Jul	ÜB	

*KW = Kalenderwoche; SW = Semesterwoche; VL = Vorlesung; ÜB = Übung; P = Praktikum

*Zwei Einführungsveranstaltungen für Praktikum: 08.05.2018 und 15.05.2018 jeweils um 17:30 Uhr am HSaF

Termine – Praktikum am Rechenzentrum

	Montag	Dienstag	Mittwoch	Donnerstag	Freitag
08:00 09:30	Pool A	Pool B Pool C			Pool A Pool B Pool C
09:45 11:15					
11:30 13:00		Pool B			Pool A Pool C
14:00 15:30	Pool I Pool K Pool L		Pool A Pool I		
15:45 17:15				Pool K	
17:30 19:00					

Gruppeneinteilung

- Einwahl durch die Studierenden in die angebotenen Termine:
ab heute 19 Uhr bis 15.05.2018 um 23:55 Uhr

- Sie vergeben 1 bis 5 Sterne zur Wahl des Wunschtermins
 - **5 Sterne** ist bevorzugter Termin
 - **Durchschnittliche** Bewertung muss mindestens **3 Sterne** betragen
 - Es muss mindestens **2 mal 5 Sterne** vergeben werden
- Via WiWi Plattform (YouSubscribe)
 - portal.wiwi.kit.edu oder wiwi.link/pit2018
- Einteilung in 3er Gruppen durch Wiwi-Portal
 - Freiwillige Gruppenbildung möglich
 - Notfalls: Bildung von 2er Gruppen
- Betreuung von ca. 4 Gruppen pro Tutor
- Tutor bleibt während der gesamten Projektphase den Gruppen zugeordnet



- Automatische Erstellung der Zugänge und Gruppen nach Anmeldung im WiWi Portal
- Alle Unterlagen und das Forum werden hier bereitgestellt
 - <https://ilias.studium.kit.edu/>
 - Eigenes Forum für jede Gruppe
 - Entsprechend 3 Mitglieder + Tutor
 - Fragen an Tutor außerhalb des Projektpraktikums
 - Diskussion über Programmteile / Vorgehensweise / Algorithmen / ...
 - Gruppenübergreifendes allgemeines Forum
 - Organisatorische Fragen
 - Fragen von allgemeinem Interesse
- Wiki kann für Informationen und Lösungsansätze genutzt werden
 - Nicht für detaillierte Lösungen verwenden
 - https://ilias.studium.kit.edu/goto.php?target=wiki_810965_Hauptseite
- Abgabe des Programms und des Berichtes
 - Siehe Folie Abgabe / Bewertung

Zeitaufteilung

- Aufbau des Projektpraktikums in verschiedene Phasen
- Vorbereitung auf die Treffen innerhalb der Gruppe
- Vorbereitung auf die Treffen mit dem Tutor (Termin)
- Anwesenheitspflicht zu den Terminen im Rechenzentrum
 - Fehlen nur aus wichtigem Grund (Abstimmung vorher mit einem Betreuer) oder mit ärztlichem Attest
- Aufgaben in der vorgegebenen Zeit nur lösbar bei
Vor- und Nachbereitung der Termine!
 - Evtl. zusätzliche Gruppentreffen erforderlich

Peer Evaluation

- Anonyme Bewertung der Studierenden einer Gruppe untereinander
- Zusätzliche Bewertung durch den Tutor
 - Verfahren der Peer Evaluation:
 - Bögen werden am 2. Termin verteilt und entsprechend am 3. Termin gesammelt
- Evaluation am 3. Termin dient zur frühen Identifizierung von Problemen in den Gruppen


TIVSEG AG
Projektpraktikum Informationstechnik
Institut für Technik der Informationsverarbeitung (ITIV)

		Selbstbeurteilung		IT-Peer-Evaluation	
		Gruppenmitglied 1	Gruppenmitglied 2	Gruppe: Name Tutor: _____ Name Teilnehmer: _____	
Interaktion mit Teammitgliedern				Dieser Peerevaluationsbogen gibt Ihnen die Möglichkeit, sich selbst und jeden Ihrer Teammitglieder im Rahmen des IT-Praktikums zu einzuschätzen. Diese Evaluation bezieht sich auf die bisherigen Praktikumsstermine. Für jeden Punkt können Sie die Bewertungsstufen der rechts aufgeführten Beschreibung entnehmen und dementsprechend das passende Feld ankreuzen. Bitte füllen Sie den Bogen vertraulich aus und geben Sie ihn bei Ihrem Tutor ab.	
				<ul style="list-style-type: none"> • Zeigt Interesse an den Ideen und dem Beitrag anderer Teammitglieder. • Fragt das Team nach Feedback und verbessert die Kommunikation. 	
				Zeigt Verhalten zwischen den oben und unten beschriebenen Punkten.	
				<ul style="list-style-type: none"> • Respektiert die Meinungen der anderen Teammitglieder. • Kommunikation mit dem Team ist in Ordnung. 	
				Zeigt Verhalten zwischen den oben und unten beschriebenen Punkten.	
Beitrag in der Teamarbeit				<ul style="list-style-type: none"> • Ist verschlossen gegenüber Ideen anderer Teammitglieder. • Vermeidet den Kontakt mit dem Team. 	
				<ul style="list-style-type: none"> • Leistet ständig gute Beiträge, welche das Team voranbringen. • Ist immer auf die Teamtreffen sehr gut vorbereitet. 	
				Zeigt Verhalten zwischen den oben und unten beschriebenen Punkten.	
				<ul style="list-style-type: none"> • Bemüht sich, sich an der Teamarbeit zu beteiligen. • Bereitet sich nur im Rahmen des Nötigen auf die Teamtreffen vor. 	
				Zeigt Verhalten zwischen den oben und unten beschriebenen Punkten.	
Qualität der Arbeit				<ul style="list-style-type: none"> • Verlangt eine unangemessene Aufteilung der Teamarbeit. • Kommt unvorbeireitet zu Teamtreffen. 	
				<ul style="list-style-type: none"> • Liefert bessere Teillösungen als erwartet und früher als zeitlich geplant. • Hält Programmierrichtlinien und im Team definierten Schnittstellen ein. 	
				Zeigt Verhalten zwischen den oben und unten beschriebenen Punkten.	
				<ul style="list-style-type: none"> • Erledigt die Teilaufgaben entsprechend den Vorgaben rechtzeitig. • Hält größtenteils Schnittstellen und Programmierrichtlinien ein. 	
				Zeigt Verhalten zwischen den oben und unten beschriebenen Punkten.	
Erfüllung der Aufgaben				<ul style="list-style-type: none"> • Liefert zu spät schlechte und unvollständige Teillösungen. • Hält keine Schnittstellen und Programmierrichtlinien ein. 	
				<ul style="list-style-type: none"> • Hat sich stark angestrengt und seine fachlichen und persönlichen Kompetenzen stark gesteigert. 	
				Zeigt Verhalten zwischen den oben und unten beschriebenen Punkten.	
				<ul style="list-style-type: none"> • Hat sich stets bemüht, sich auf fachlicher und persönlicher Ebene weiterzuentwickeln. 	
				Zeigt Verhalten zwischen den oben und unten beschriebenen Punkten.	
Gesamtbeurteilung				<ul style="list-style-type: none"> • Hat kein Interesse gezeigt, sich persönlich weiterzuentwickeln und seine fachlichen Kenntnisse zu erweitern. 	
				Sehr gut	
				Gut	
				Befriedigend	
				Ausreichend Mangelhaft	



Rückblick



Rückblick



Bewertung

- Bewertung der abgegebenen Dokumente nach Kriterien aus dem Lastenheft und ILIAS
- Es müssen 100% der Spezifikation / Aufgabenstellung erfüllt werden, um das Praktikum erfolgreich abzugeben
 - Verschiedene Lösungswege möglich sein
 - Programmcode dokumentiert werden
- Programmierfähigkeiten nachweisbar sein
 - Einzelwertungen der simulierten Testfahrt je nach Implementierung sind möglich.
 - Vorführung am letzten Termin auf dem TivSeg ist freiwillig.
- Kein Copy- & Paste von anderen Gruppen zugelassen
- Keine separate schriftliche Prüfung, aber der Inhalt in IT-Modul Klausur geprüft

**Wichtig: Im Studierendenportal für PIT an-/ abmelden:
bis 03.06.2018 – entspricht bis dem Ende der 1. PIT-Woche**

- Abgabedokumente müssen fristgerecht in ILIAS hochgeladen werden
 - Abgabe des Pflichtenhefts inklusive Zeitplan: Spätestens bis 7. Tag nach dem 1. Praktikumstermin
 - Abgabe der funktionierenden HAL: Spätestens bis 7. Tag nach dem 4. Praktikumstermin
 - Abgabe der Gesamtsoftware: Spätestens bis 7. Tag nach dem 6. Praktikumstermin
 - Abgabe der Projektdokumentation: Spätestens bis 7. Tag nach dem 6. Praktikumstermin
 - Probefahrt in Woche 7 ist freiwillig
- Detail zu den Liefergegenständen Folie 40ff
- ILIAS sperrt das Hochladen automatisch, Fristen sind auf ILIAS hinterlegt
 - Bitte melden Sie sich bei Problemen rechtzeitig (vor Fristende)
- Vorlagen für Dokumentation und Zeitplanung sind auf ILIAS verfügbar
- Hochladen der Header- & Programmdateien (nur *.h und *.cpp) und der Berichte (nur *.pdf – kein *.doc) auf ILIAS

- Zuordnung der Gruppen
- Betreuung durch Tutoren
- Durchführung im SCC
- Vor- und Nachbereitung zu Hause
- Anmeldung auf WiWi Lernplattform



Auftraggeber und Problemstellung (I)



■ TivSeg AG

- Herstellung von TivSegs™
 - selbstbalancierendes einachsiges Beförderungsmittel
- Mechanischer Aufbau ist bereits vorhanden
- Regelalgorithmus ist bereits vorhanden

- Ansteuerung der Hardware ist nicht vorhanden
 - Ansteuerung der Motoren
 - Auslesen der Neigungssensoren und Beschleunigungssensoren
 - Auslesen des Lenkwinkels





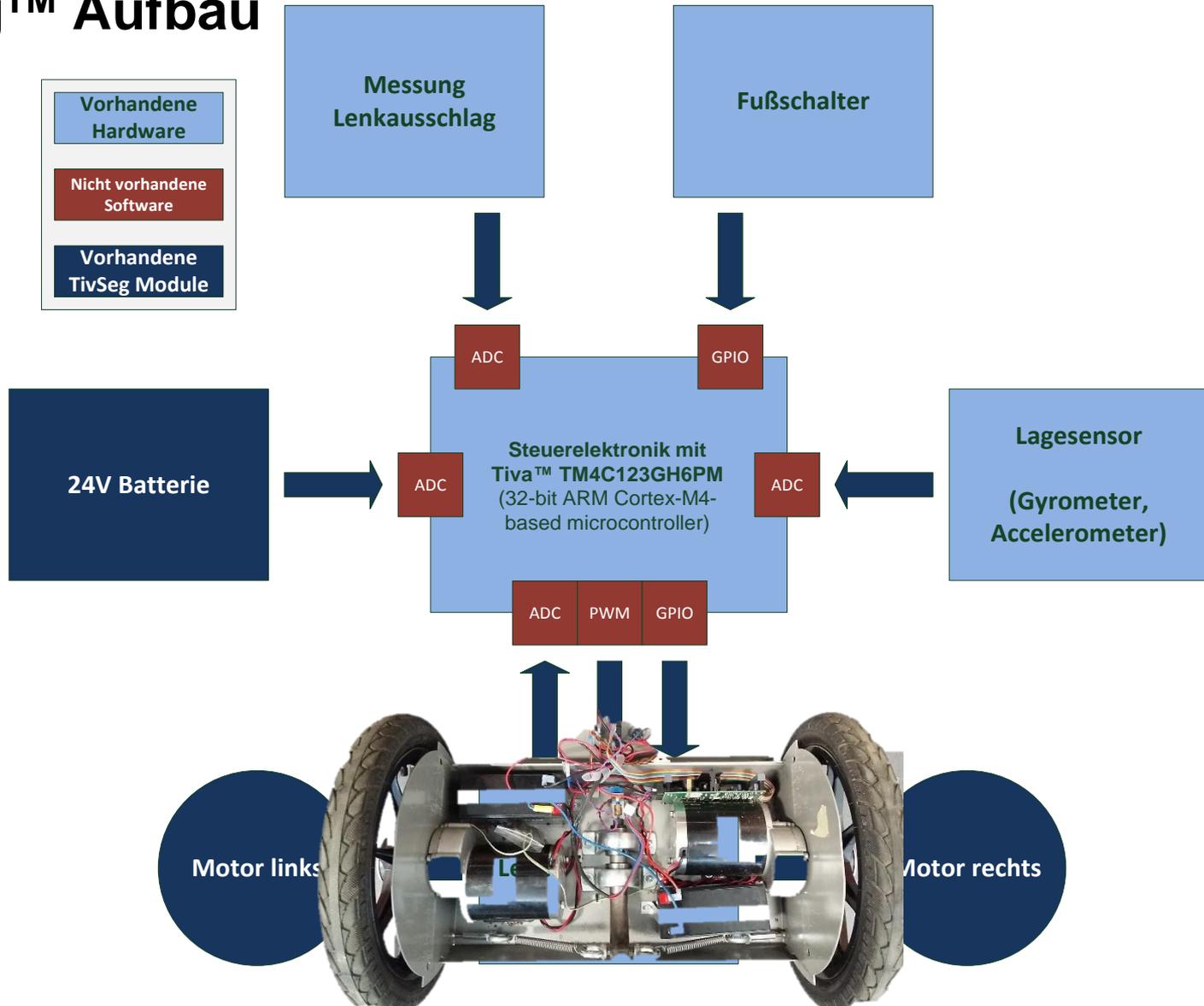
Auftraggeber und Problemstellung (II)

- Kundenauftrag zur Entwicklung einer neuen Software zur Steuerung eines TivSeg™
- Sie werden dringend zur Verstärkung des Teams benötigt
- Bitte halten Sie sich an Vorgaben des Kunden (Lastenheft! Anforderungen aus Sicht des Auftraggebers)
 - Download mit Passwort: **PITSS18**
 - http://www.itiv.kit.edu/60_6070.php
 - Richtlinien zur Formatierung des Quelltextes
 - Bitte streng befolgen
- Regler ist bereits vorhanden
 - Allerdings wird eine Ansteuerung des Reglers benötigt
- Dokumentation sollte mit Latex erstellt werden
 - Entsprechende Vorlagen sind vorhanden

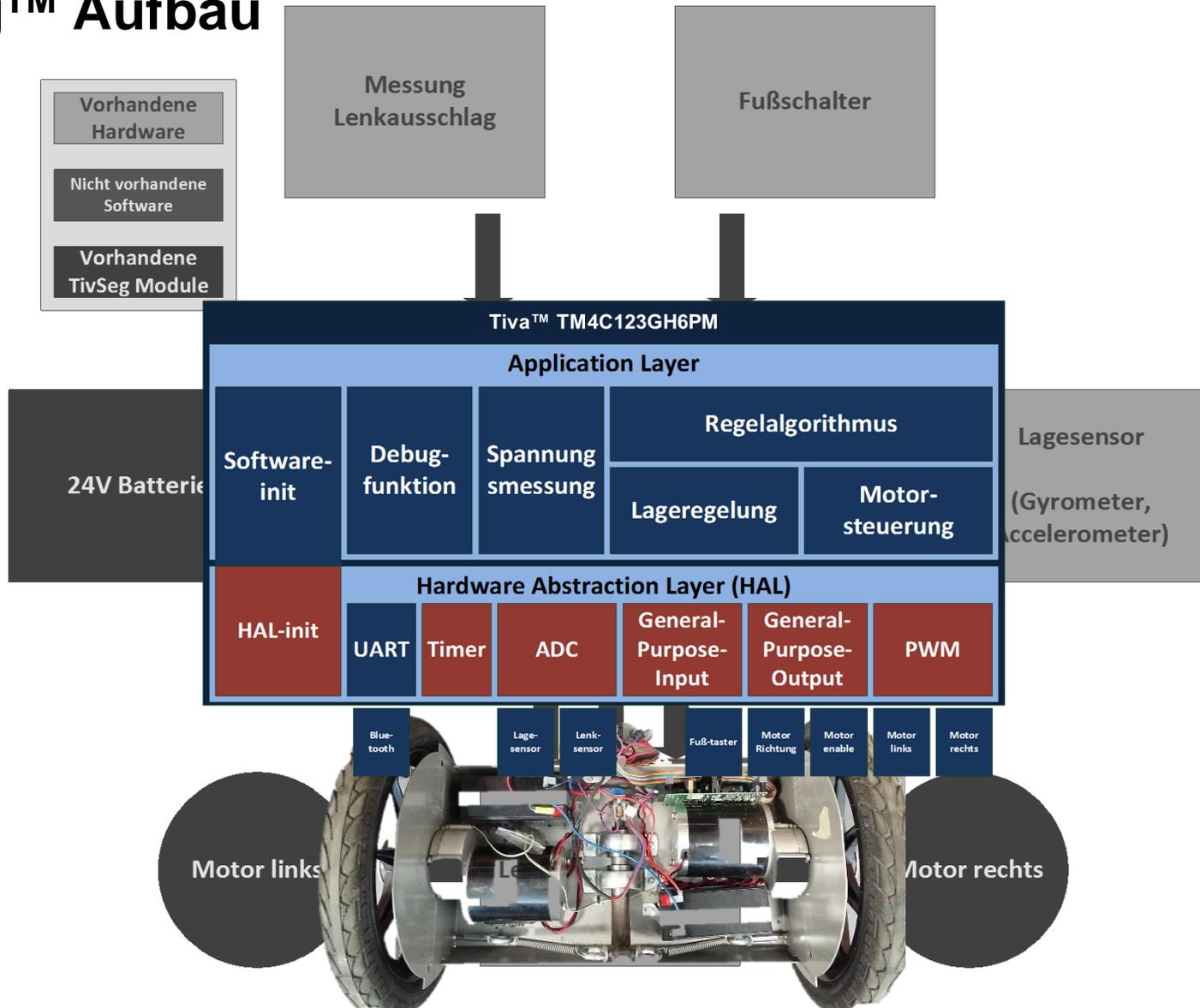
Live fahren



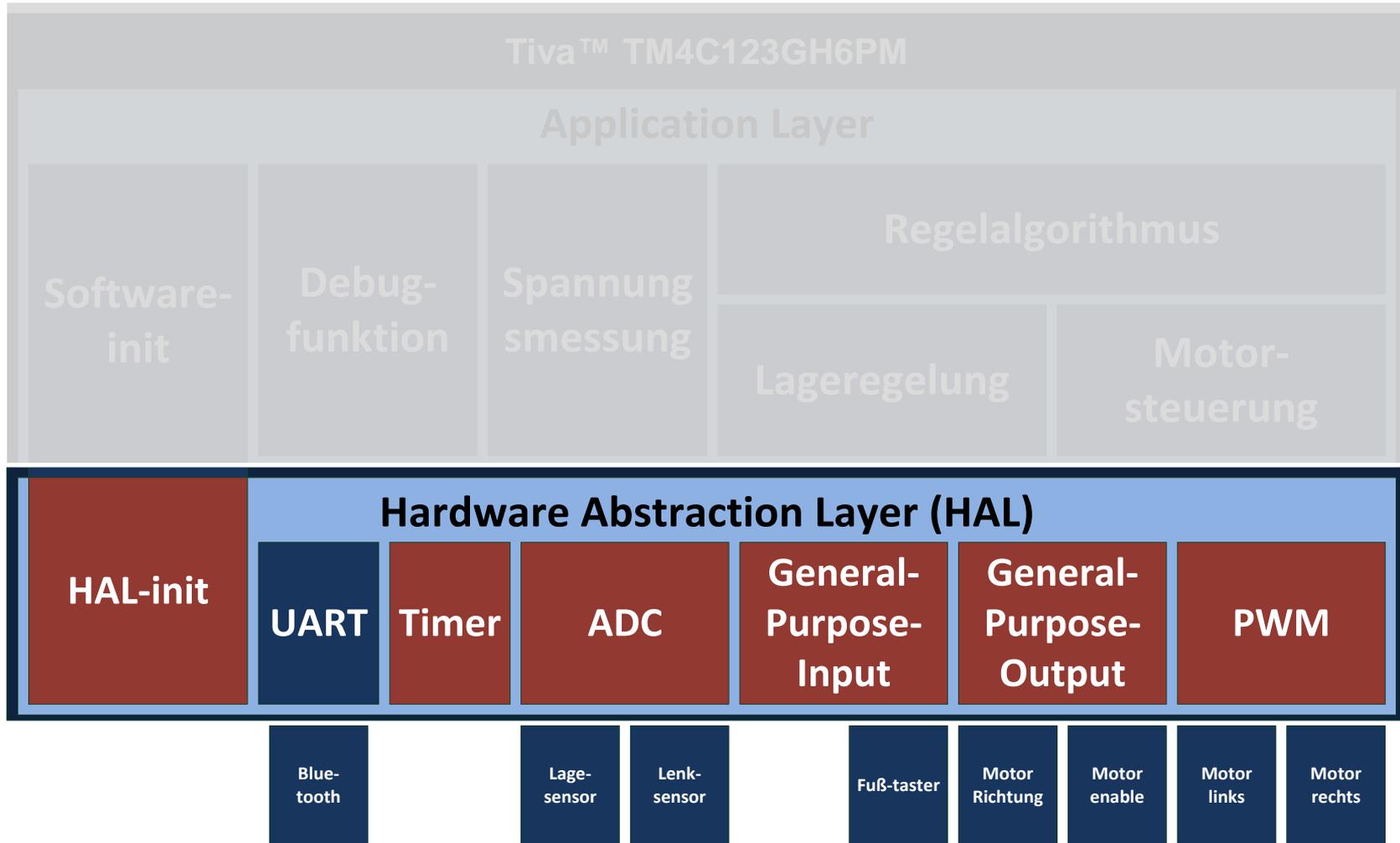
TivSeg™ Aufbau



TivSeg™ Aufbau

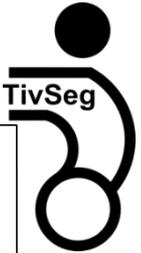


TivSeg™ Gesamtübersicht



Aufbau eines von Neumann Rechners

Detaillierte Sichtweise (bekannt aus Vorlesung)



Steuerwerk

- BR: Befehlsregister
 - BRC: Codeteil (Opcode)
 - BRA: Adressteil
- ST: Register für Steuerbits
- BZ: Befehlszähler
 - BZR: Befehlszählerregister

Speicherwerk

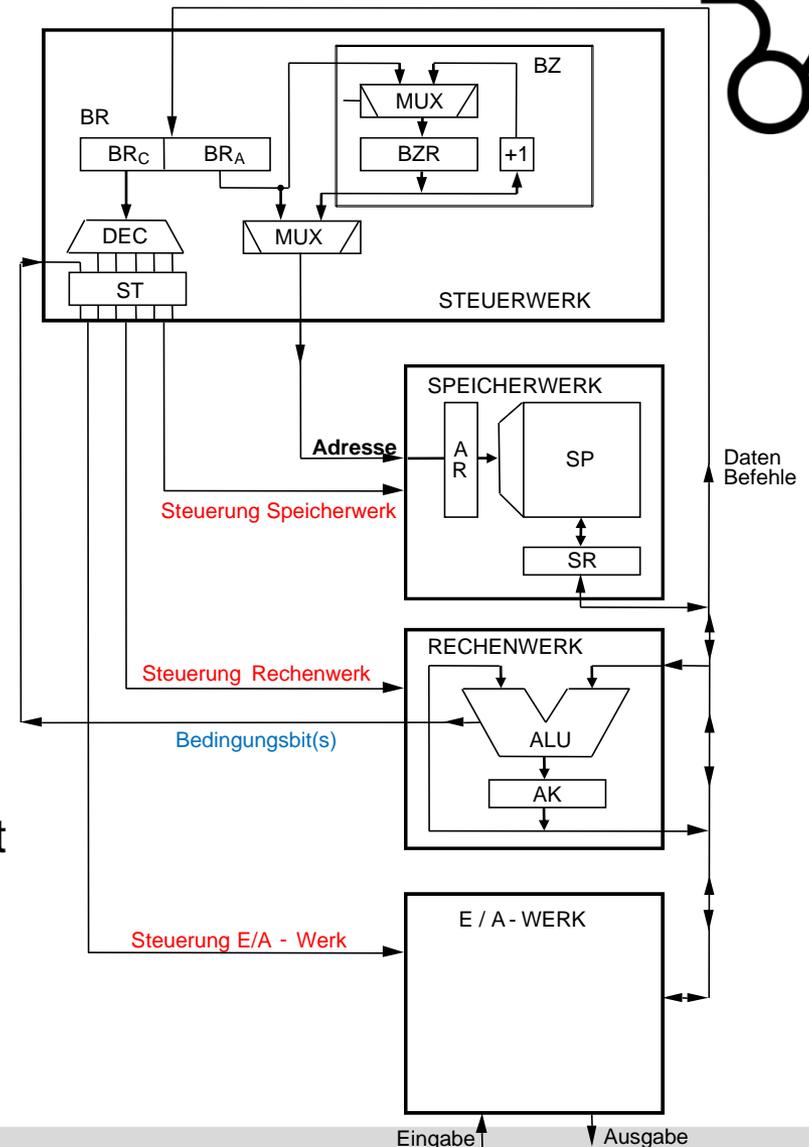
- AR: Adresszähler
- SP: Speicher
- SR: Speicherregister

Rechenwerk

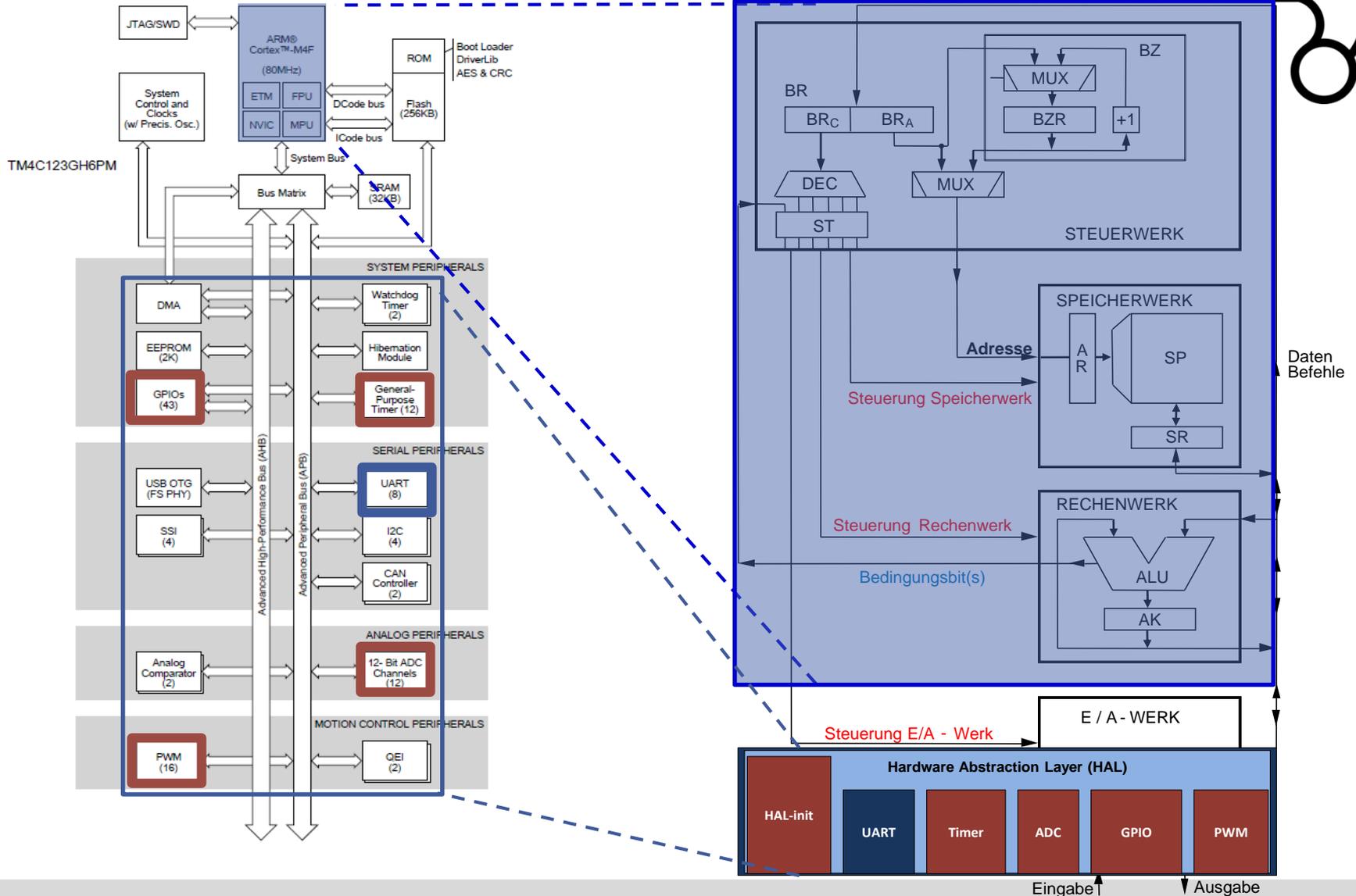
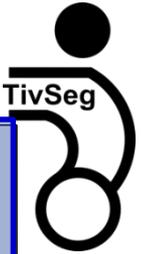
- ALU: Arithmetische-, logische Einheit
- AK: Akkumulator

E/A-Werk

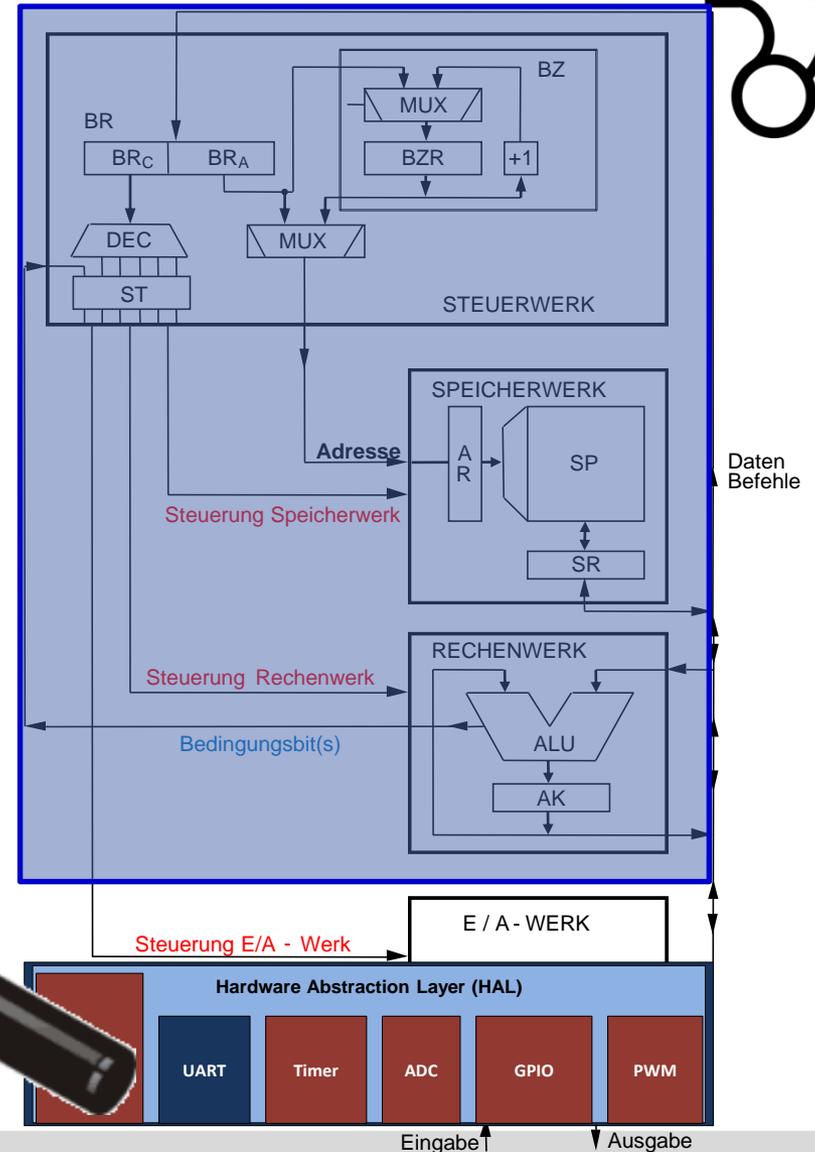
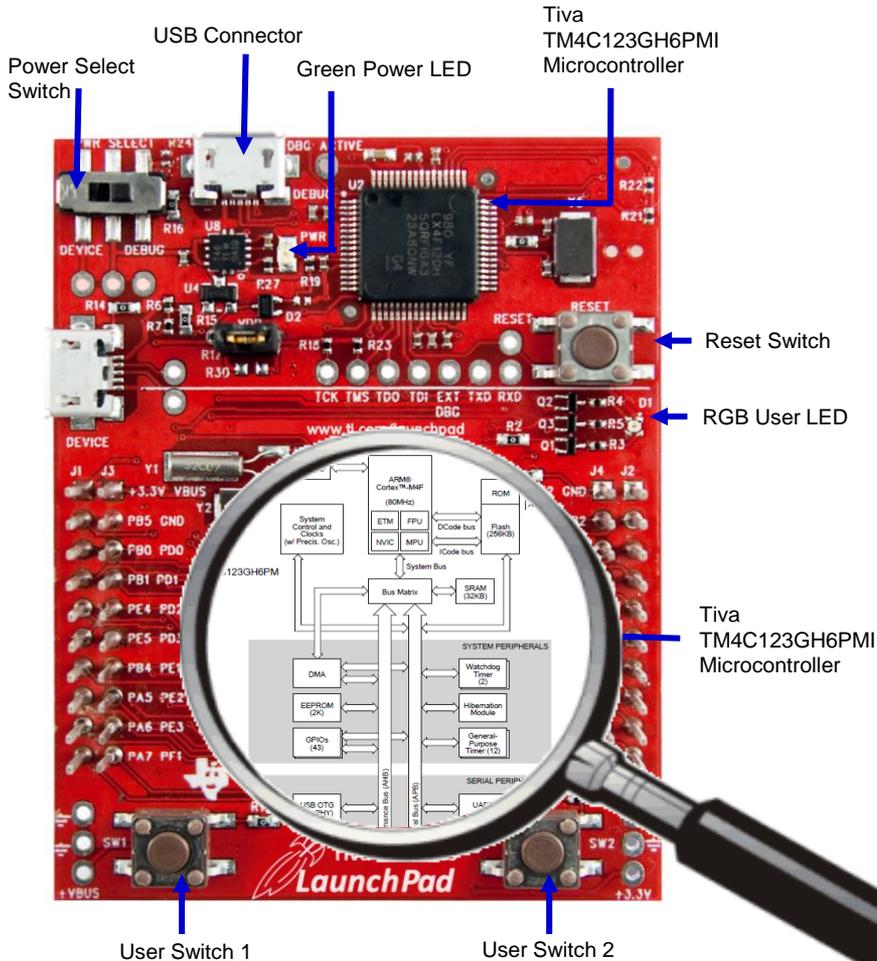
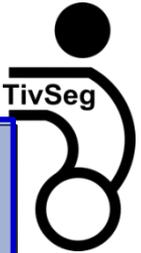
- E/A: Eingabe/Ausgabe



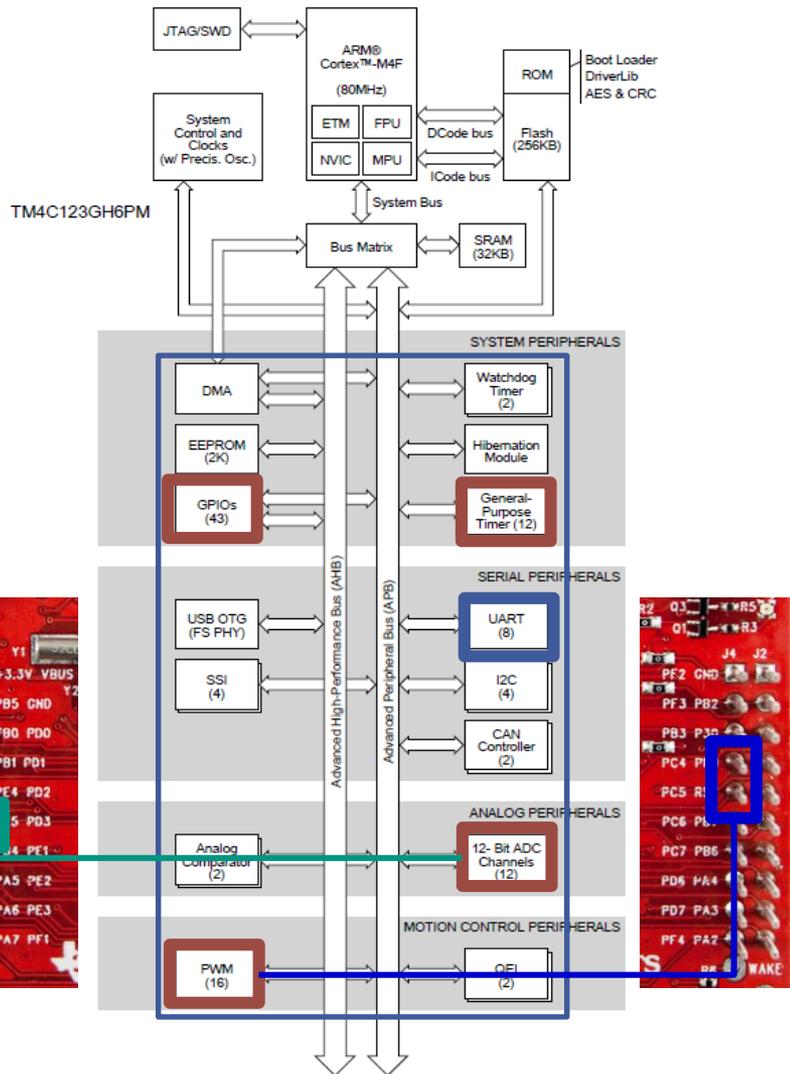
Einordnung in allgemeine Rechnerarchitektur



Einordnung in allgemeine Rechnerarchitektur



Einordnung in allgemeine Rechnerarchitektur



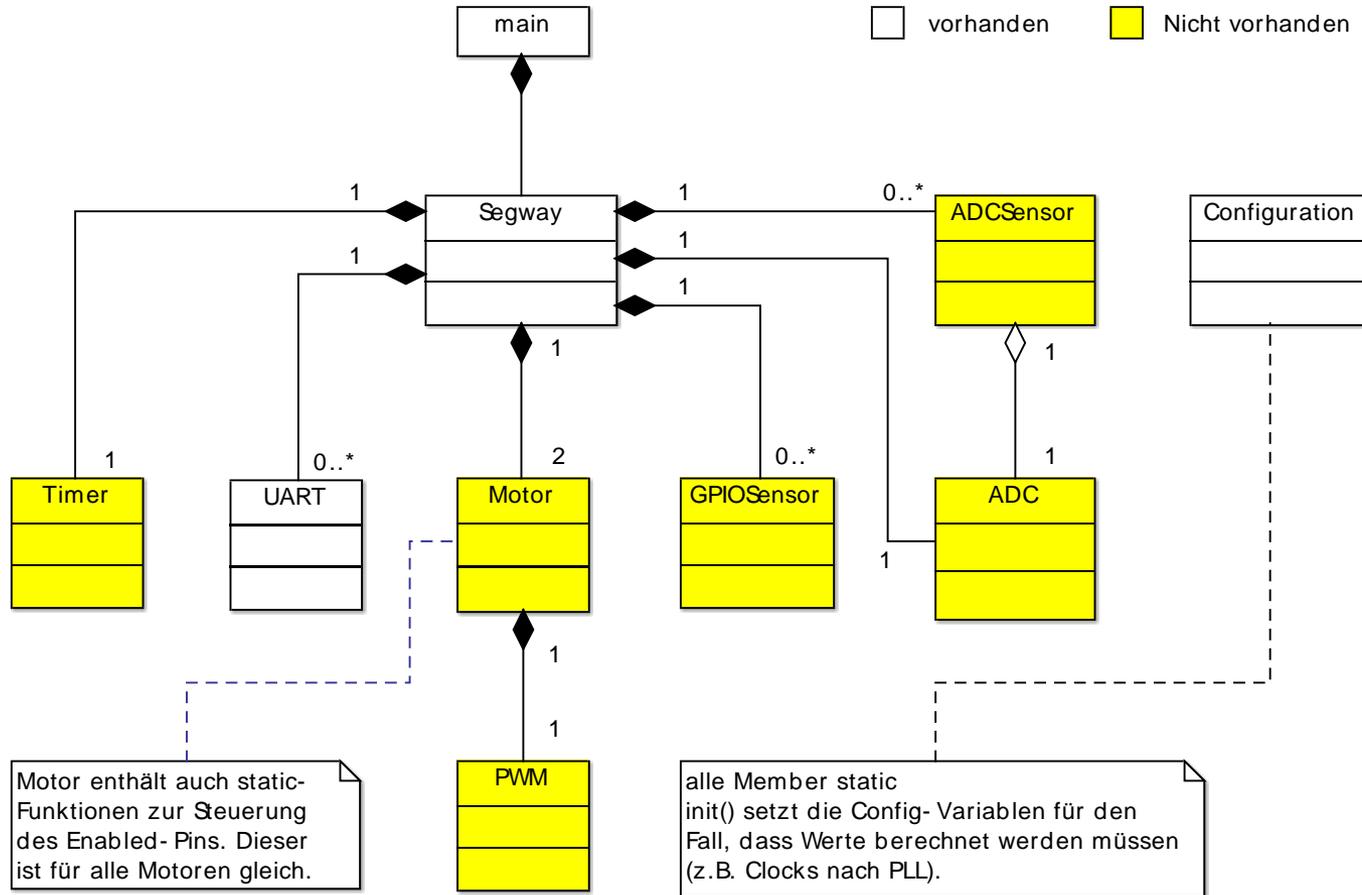
Anhang A

Pinbelegung des TM4C123G

Pin	Verwendung	Pin	Verwendung
PB05	Not Connected (NC)	PB2	NC
PB0	UART: RX	PE0	Rechter Onboard-Button
PB1	UART: TX	RST	Reset Button
PE4	Strommessung linker Motor (AIN9)	PB7	SPI-MOSI
PE5	Strommessung rechter Motor (AIN8)	PB6	SPI-MISO
PB4	NC	PA4	NC
PA5	NC	PA3	NC
PA6	I2C-Bus: SCL	PA2	Fußschalter
PA7	I2C-Bus: SDA	PF2	Blaue Onboard LED
PD0	Drehratensensor 2 (AIN7)	PF3	Rote Onboard LED
PD1	Drehratensensor 1 (AIN6)	PB3	NC
PD2	Spannungsteiler Batterie (AIN5)	PC4	Linker Motor PWM
PD3	Potenzimeter Lenkausschlag (AIN4)	PC5	Rechter Motor PWM
PE1	Accelerometer X (AIN2)	PC6	Rechter Motor Richtungspin
PE2	Gyrometer Referenzspannung (AIN1)	PC7	Linker Motor Richtungspin
PE3	Gyrometer Y (AIN0)	PD6	Motoren Enable-Pin
PF1	Rote Onboard LED	PD7	NC
PF4	Linker Onboard-Button		

Tabelle A.1: Pinbelegung des TM4C123G

TivSeg™ Software Spezifikation

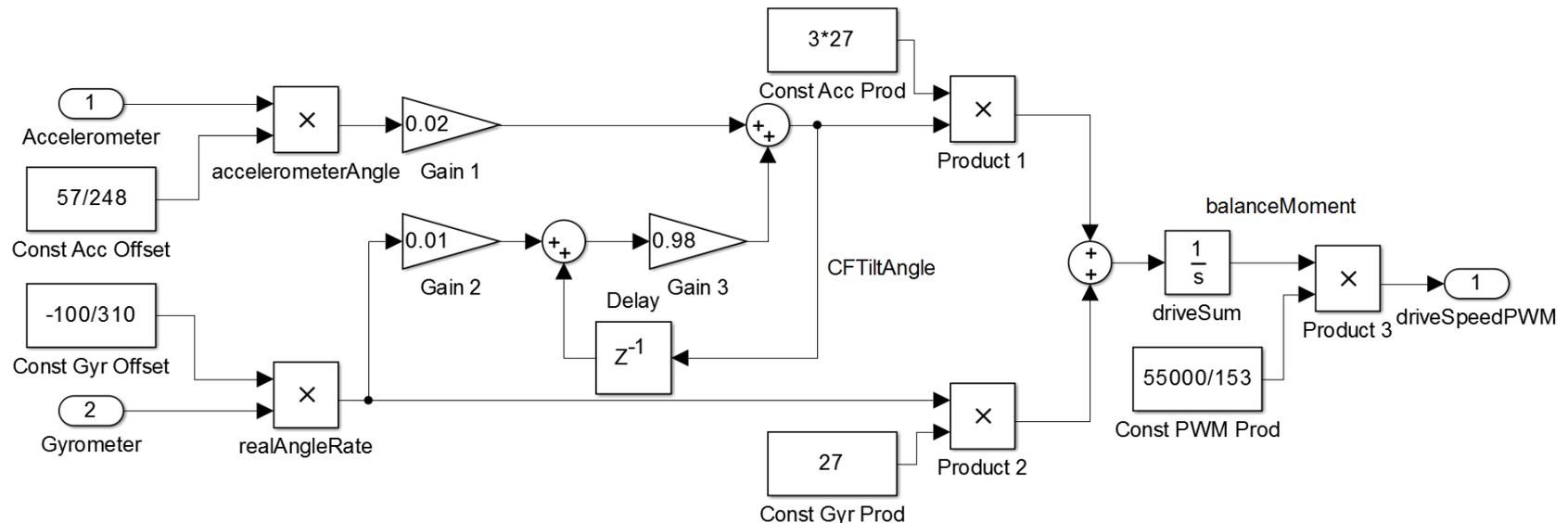


Beschreibung der vorhandenen Regelung

- Matlab Modell des Regelalgorithmus konnte erworben werden

- PI-Regler

- Eingänge: Gyrometer und Beschleunigungssensor
- Rückkopplung in Mitkopplung
- Post-Scaler zur Verfeinerung der Ansteuerung der Motoren



Modellierung der Regelung mit Matlab



Steuerung

start stop

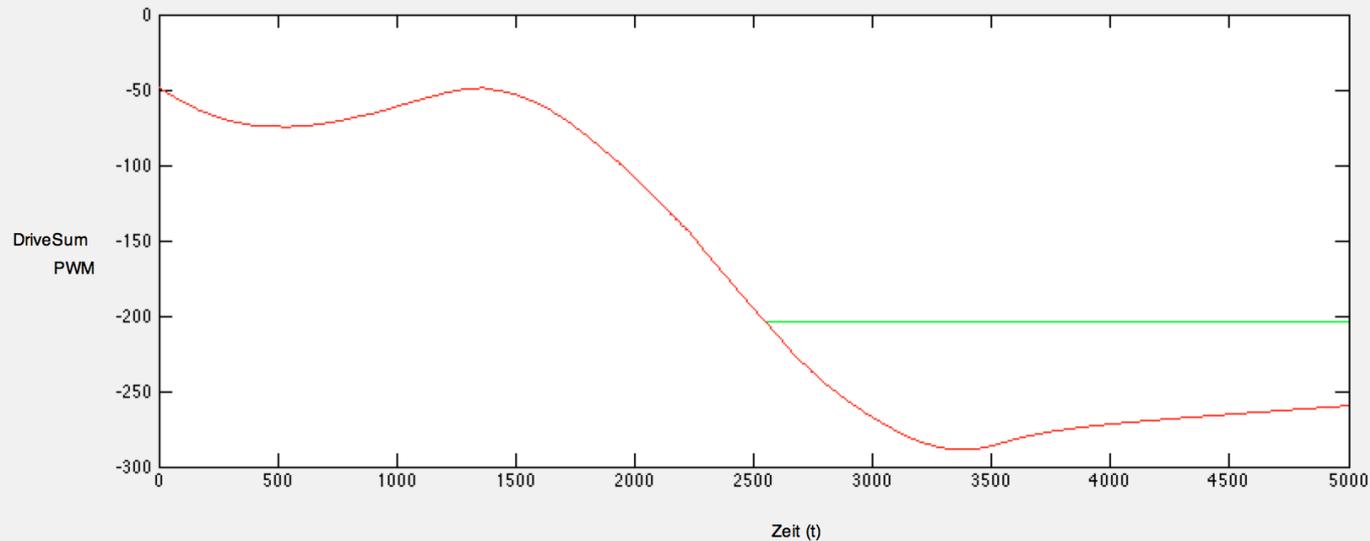
0

Stehen Bleiben



PWM

DriveSum

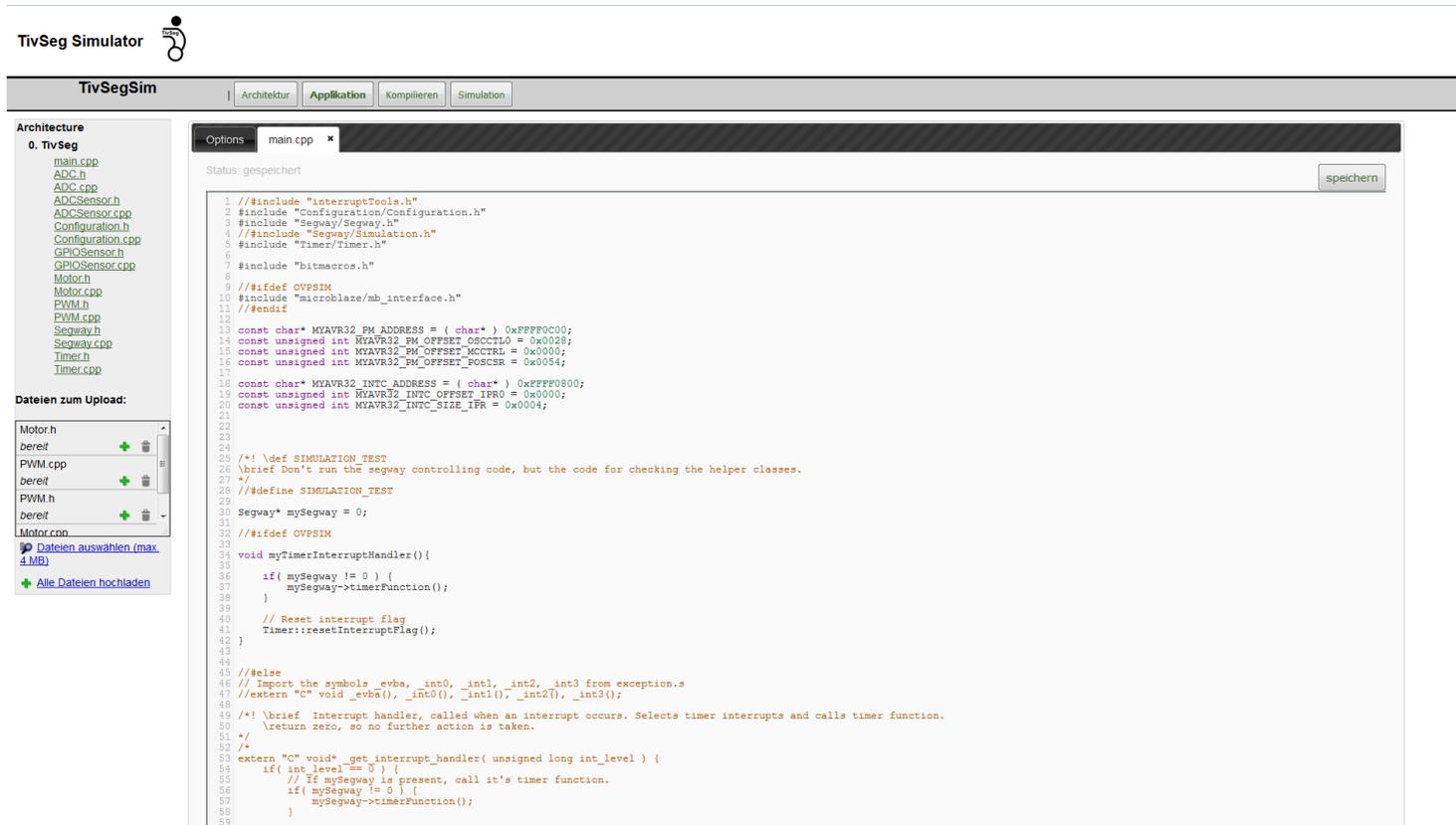


Start der virtuellen Maschinen in SCC-Pools

- Im Bootloader den Eintrag „**BW-Lehrpool**“ auswählen
- Login nach Aufforderung auf der Konsole über **u-Account**
- Im „**bwLehrpool VMChooser**“ die virtuelle Maschine „**ITIV-Praktikum**“ per Doppelklick auswählen
- Windows startet mit Administratorrechten
- **Datenspeicherung nur auf Home-Netzlaufwerk oder USB-Stick!**
 - Sämtlichen Daten auf der lokalen Platte gehen durch Reboot der virtuellen Maschine verloren!

Test des Codes auf OVP

- <http://simplify.itiv.kit.edu/tivseg-sim>
 - Zusammensetzung von Username und Passwort im ILIAS-Wiki
 - Wird nach Gruppeneinteilung bekanntgegeben



The screenshot shows the TivSeg Simulator interface. On the left, there is a file explorer showing the project structure under '0. TivSeg', including files like main.cpp, ADC.h, ADCSensor.cpp, Configuration.h, GPISensor.h, GPISensor.cpp, Motor.h, Motor.cpp, PWM.h, PWM.cpp, Segway.h, Segway.cpp, Timer.h, and Timer.cpp. Below the file explorer is a section for uploading files. The main area is a code editor for 'main.cpp' with the following code:

```
Options | main.cpp *
Status: gespeichert
speichern

1 // #include "interruptTools.h"
2 #include "Configuration/Configuration.h"
3 #include "Segway/Segway.h"
4 // #include "Segway/Simulation.h"
5 #include "Timer/Timer.h"
6
7 #include "bitmacros.h"
8
9 // #ifdef OVPSIM
10 #include "microblaze/mb_interface.h"
11 // #endif
12
13 const char* MYAVR32_PM_ADDRESS = (char*) 0xFFFF0C00;
14 const unsigned int MYAVR32_PM_OFFSET_OSCCTRL = 0x0028;
15 const unsigned int MYAVR32_PM_OFFSET_MCCTRL = 0x0000;
16 const unsigned int MYAVR32_PM_OFFSET_POSCSR = 0x0054;
17
18 const char* MYAVR32_INTC_ADDRESS = (char*) 0xFFFF0800;
19 const unsigned int MYAVR32_INTC_OFFSET_IPRO = 0x0000;
20 const unsigned int MYAVR32_INTC_SIZE_IPR = 0x0004;
21
22
23
24
25 /*! \def SIMULATION_TEST
26 \brief Don't run the segway controlling code, but the code for checking the helper classes.
27 */
28 // #define SIMULATION_TEST
29
30 Segway* mySegway = 0;
31
32 // #ifdef OVPSIM
33
34 void myTimerInterruptHandler() {
35     if( mySegway != 0 ) {
36         mySegway->timerFunction();
37     }
38 }
39
40 // Reset interrupt flag
41 Timer::resetInterruptFlag();
42 }
43
44
45 // #else
46 // Import the symbols evbs, _int0, _int1, _int2, _int3 from exception.s
47 // extern "C" void _evb0(), _int0(), _int1(), _int2(), _int3();
48
49 /*! \brief Interrupt handler, called when an interrupt occurs. Selects timer interrupts and calls timer function.
50 \return zero, so no further action is taken.
51 */
52 /*
53 extern "C" void* _get_interrupt_handler( unsigned long int_level ) {
54     if( int_level == 0 ) {
55         // If mySegway is present, call it's timer function.
56         if( mySegway != 0 ) {
57             mySegway->timerFunction();
58         }
59     }
60 }
```


- Willkommen im Entwicklungsteam der TivSeg AG
- Planung und Umsetzung, sowie Testen und Erproben des TivSeg™
- Vorgaben des Kunden müssen eingehalten werden
- Aufbau des TivSeg™

A large, stylized question mark with a thick green outline and a white fill, positioned to the right of the list items.

Fragen?

**Wichtig: Im Studierendenportal für PIT an-/ abmelden:
bis 03.06.2018 – entspricht bis dem Ende der 1. PIT-Woche**

Was der Kunde sich wünscht



Was der Kunde sich wünscht



Was der Kunde
erklärte

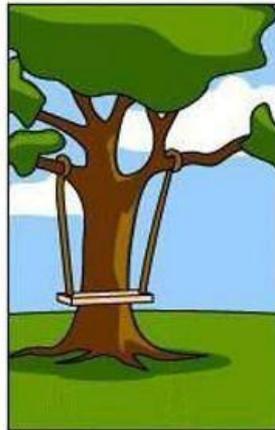


Was der Kunde wirklich
gebraucht hätte

Was der Kunde sich wünscht



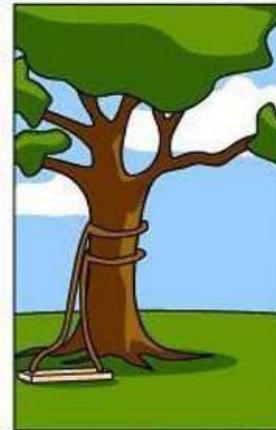
Was der Kunde erklärte



Was der Projektleiter verstand



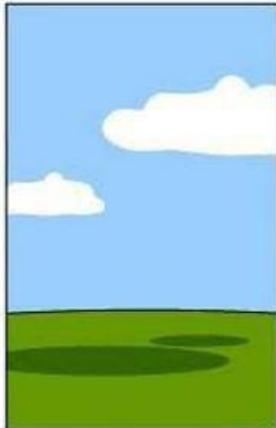
Wie es der Analytiker entwarf



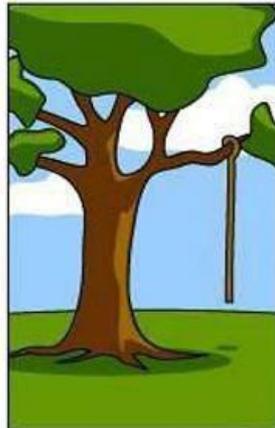
Was der Programmierer programmierte



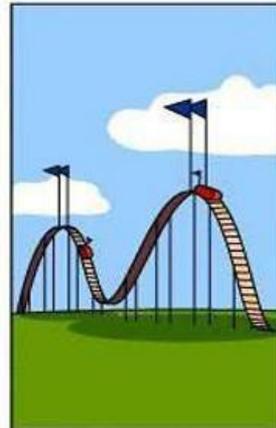
Was der Berater definierte



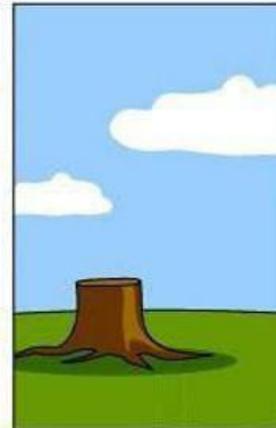
Wie das Projekt dokumentiert wurde



Was installiert wurde



Was dem Kunden in Rechnung gestellt wurde



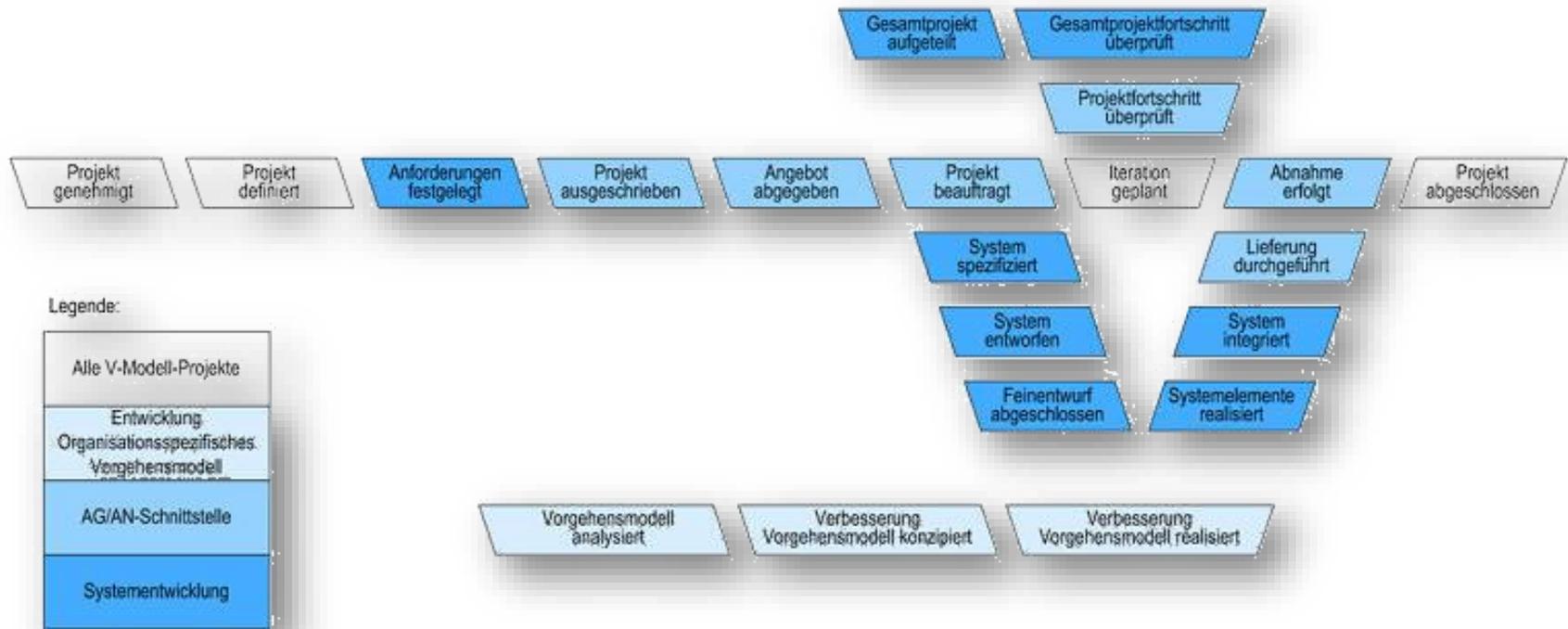
Wie es gewartet wurde



Was der Kunde wirklich gebraucht hätte

Vom Problem zum Algorithmus

- Für große Aufgabenstellungen s. Vorgehensmodelle (V-Modell, Spiralmodell etc.)
 - in der Vorlesung „Systems and Software Engineering“



Anwendungsgebiete / -fälle

- Bei diversen großen Industrie-Firmen (z. B. T-Systems, Deutsche Post, LBS, Quelle-Versicherung)
- Generell geeignet für alle Unternehmen, insbesondere auch für kleine und mittelständische Unternehmen
- Das V-Modell XT für alle Ressorts der Bundesbehörden anzuwenden, auch ohne einen weiteren Erlass
- Beispiel: Klassisches Fahrzeugprojekt



Beispiele für Grenzen

- Komplette Neu-Entwicklung (disruptive Technologie)
 - Keine Erfahrungswerte
 - Keine Kundeneinschätzung
 - Keine Partnerlandschaft
 - Kurzfristige Änderungen
 - Unsicherheit über Anforderungen
- Beispiele der letzten Zeit:
 - A-Klasse der 1. Generation
 - Frontmotor
 - Neues Marktsegment
 - Elektrische Fahrzeuge (1)
- Schnelle erste Ergebnisse
 - Designstudie (2)
- Neue HMI-Komponenten
 - Drehdrücksteller
 - Neues Kombi-Design (3)



Vorgaben

- Spezifikation in natürlicher Sprache
 - Zum Teil mit UML Klassendiagramm, Schaubilder, Tabellen, ...
- Objektorientierter Aufbau des Programms
 - Aufteilung der Problemstellung
 - Übersichtlicher Aufbau
- Aufteilung der Aufgabenstellung in Blöcke
- Eigene Vorgehensweise – Abstimmung in der Gruppe, ggf. mit Tutor

Projektplanung (inkl. Abgabetermine aus Folie 12)

1. Termin

- Planung

Liefergegenstand

- Pflichtenheft

2. Termin

- Umsetzung 1

3. Termin

- Umsetzung 2

4. Termin

- Umsetzung 3

Liefergegenstand

- Funktionierende HAL

5. Termin

- Test der Einzelmodule

6. Termin

- Abschlusstest und Dokumentation

Liefergegenstände

- Projektdokumentation + Software

7. Termin

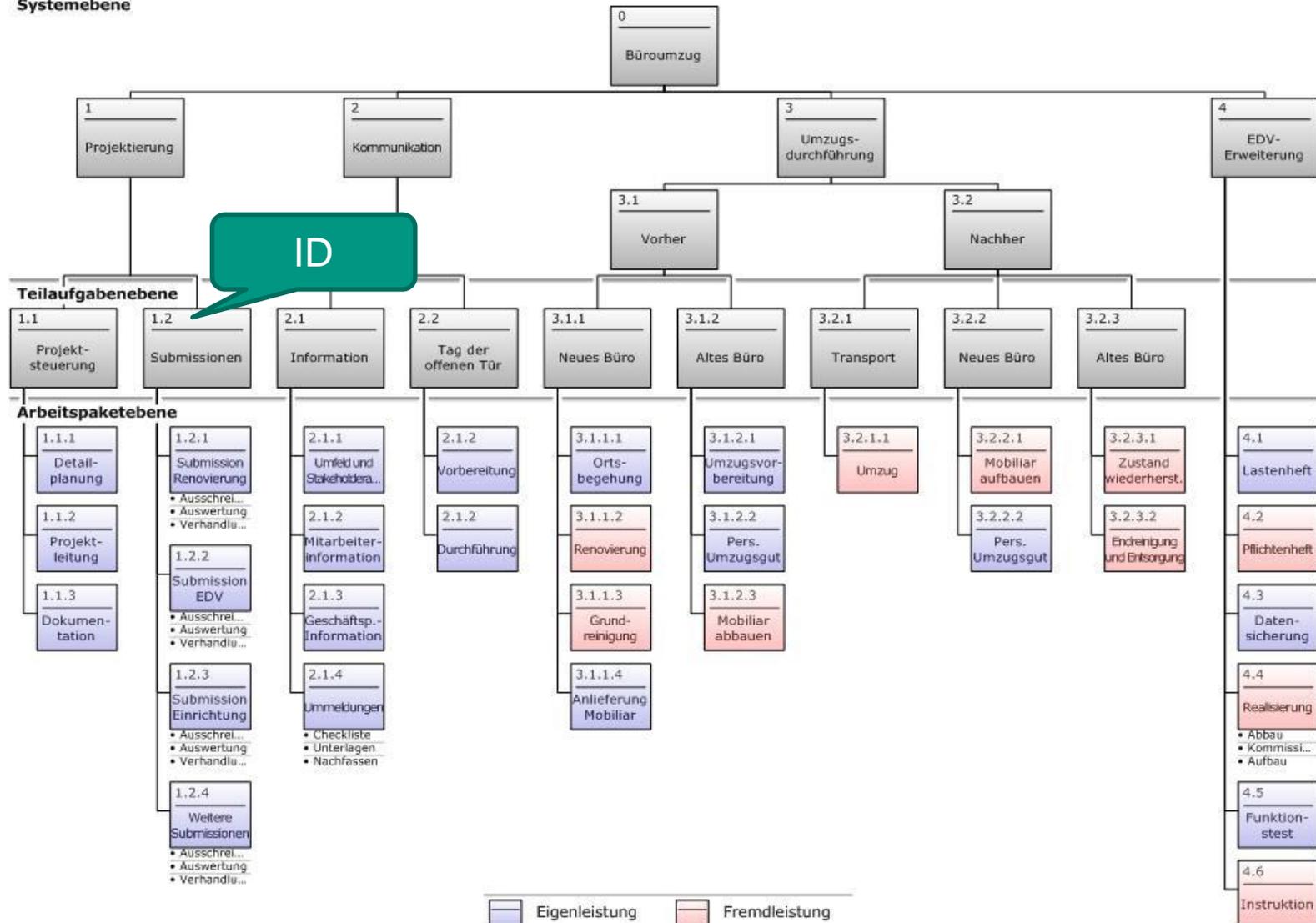
- Freiwillige Probefahrt

1. Termin: Planung

- Knackpunkte identifizieren (nicht lösen)
- Aufgabenverteilung
 - Faire Aufgabenverteilung (nicht nach Vorwissen)
 - Jeder sollte alle Teile kennen
- Aufbau des Programms
 - Ergänzung des vorgegebenen UML-Klassendiagramms
 - Schnittstellen u. a. über Übergabe- und Rückgabeparameter
- Liefergegenstände:
 - Pflichtenheft (Detaillierung der Kundenanforderungen aus Sicht des Auftragnehmers, s. Dokumentation)
 - Projektstrukturplan
 - Beschreibung der TivSeg™-Funktionsweise (Ablaufdiagramm)
 - Beschreibung der Schnittstellen mit Klassendiagramm
 - Zeitplan (Gantt-Chart)
 - Planung der einzelnen Arbeitspakete
 - Festlegen von internen Deadlines / Meilensteinen

Projektstrukturplan

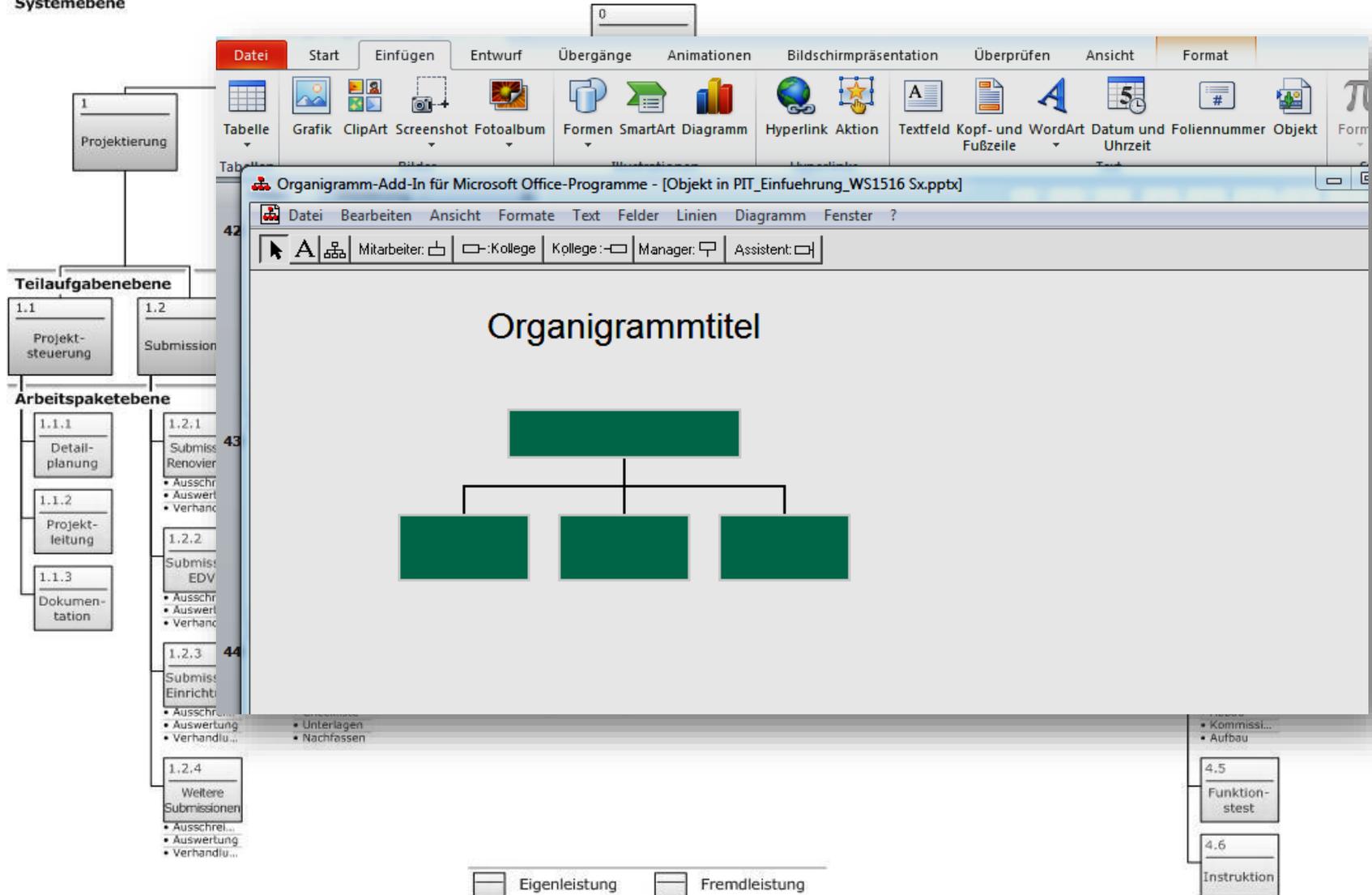
Systemebene



Eigenleistung
 Fremdleistung

Projektstrukturplan

Systemebene



The image displays a software interface for creating a Project Structure Plan (PSP) and an Organizational Chart (Organigramm).

Systemebene (Left):

- 1 Projektierung

Teilaufgabenebene (Left):

- 1.1 Projektsteuerung
- 1.2 Submission

Arbeitspaketebene (Left):

- 1.1.1 Detailplanung
- 1.1.2 Projektleitung
- 1.1.3 Dokumentation
- 1.2.1 Submiss Renover
 - Ausschr
 - Auswert
 - Verhandl...
- 1.2.2 Submiss EDV
 - Ausschr
 - Auswert
 - Verhandl...
- 1.2.3 Submiss Einricht
 - Ausschr...
 - Auswertung
 - Verhandlu...
- 1.2.4 Weitere Submissionen
 - Ausschrel...
 - Auswertung
 - Verhandlu...

Organigramm-Add-In für Microsoft Office-Programme - [Objekt in PIT_Einfuehrung_WS1516 Sx.pptx] (Right Window):

Organigrammtitel

```

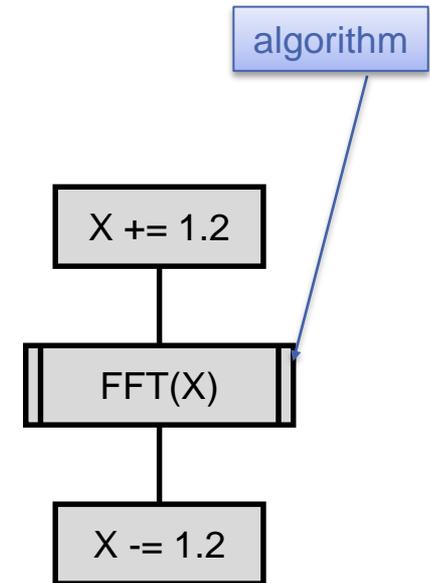
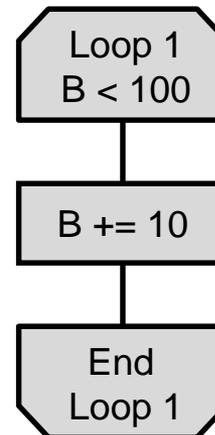
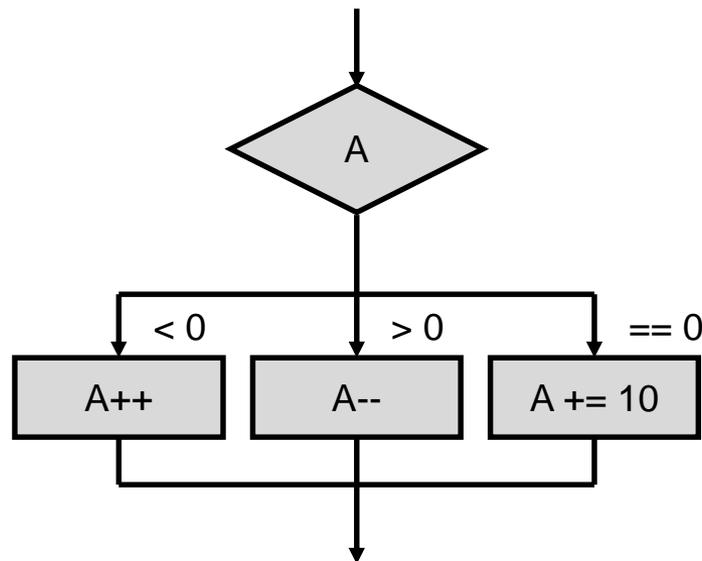
graph TD
    A[ ] --- B[ ]
    A --- C[ ]
    A --- D[ ]
    
```

Legend (Bottom):

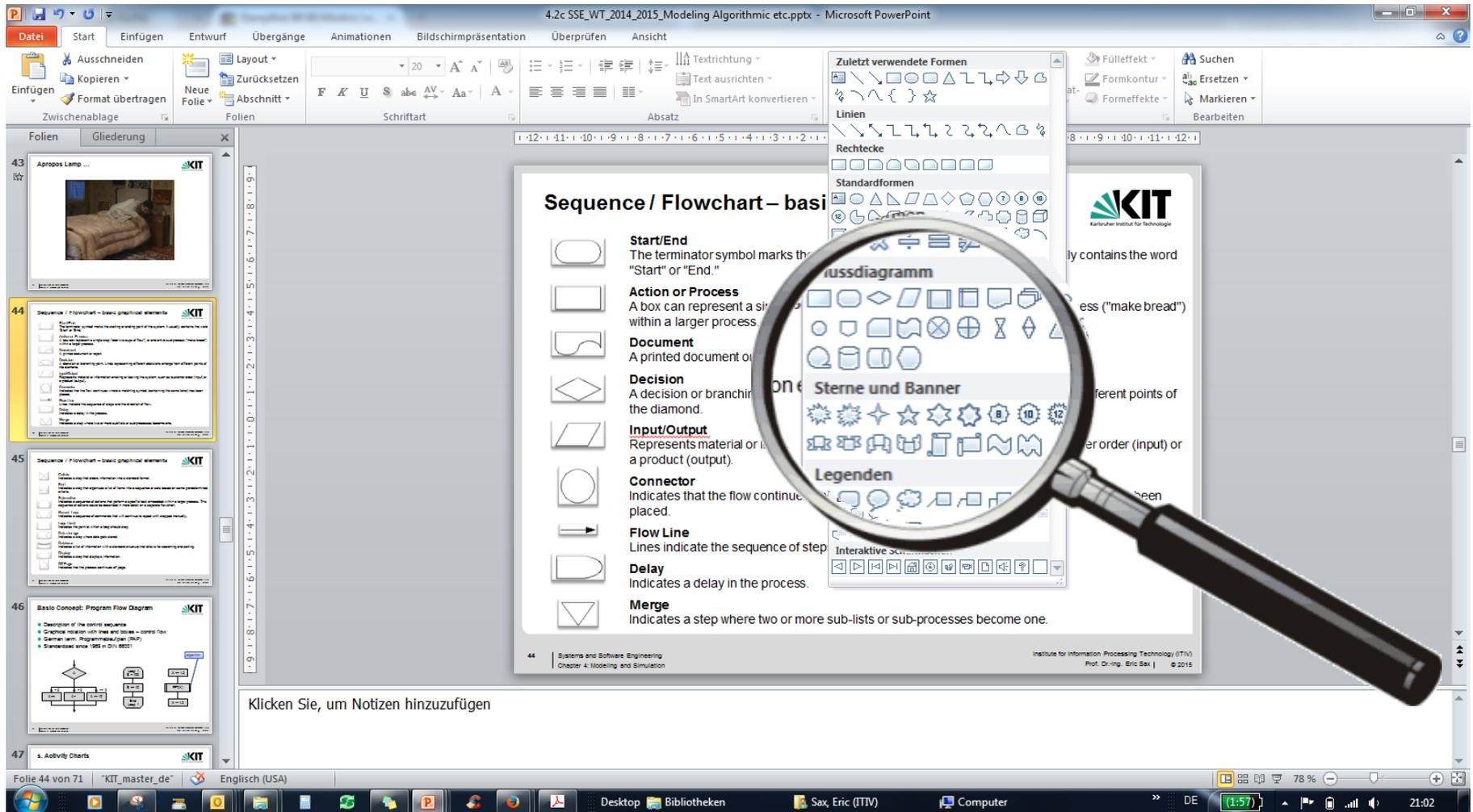
- Eigenleistung
- Fremdleistung

Basic Concept: Program Flow Diagram

- Description of the control sequence
- Graphical notation with lines and boxes – control flow
- Standardized in DIN 66001



Flow Diagram „Editor“



Sequence / Flowchart – basi

- Start/End**
The terminator symbol marks the "Start" or "End."
- Action or Process**
A box can represent a simple process within a larger process.
- Document**
A printed document or a document.
- Decision**
A decision or branching point on the diamond.
- Input/Output**
Represents material or information entering or leaving a process (input or output).
- Connector**
Indicates that the flow continues from another connector to be placed.
- Flow Line**
Lines indicate the sequence of steps.
- Delay**
Indicates a delay in the process.
- Merge**
Indicates a step where two or more sub-lists or sub-processes become one.

Zuletzt verwendete Formen

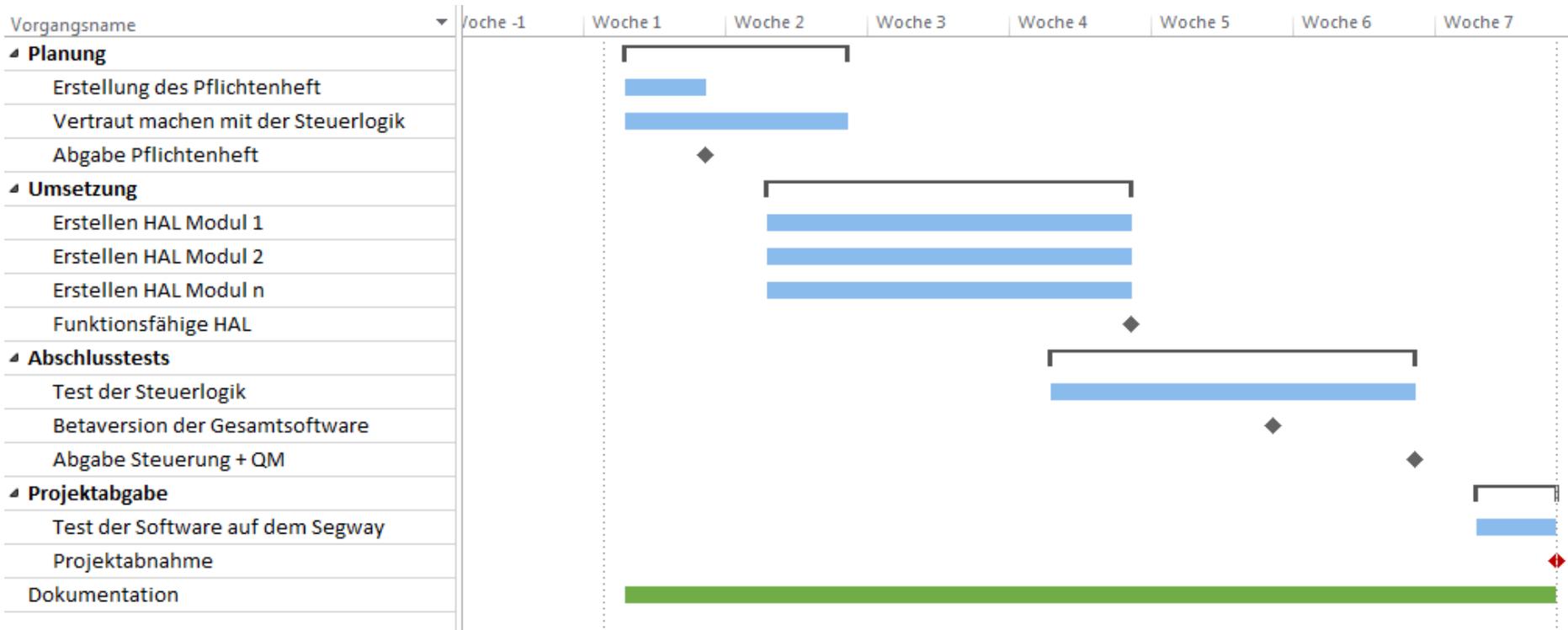
- Linien
- Rechtecke
- Standardformen
- Sterne und Banner
- Legenden

Klicken Sie, um Notizen hinzuzufügen

Folie 44 von 71 "KIT_master_de" Englisch (USA) 78% 21:02

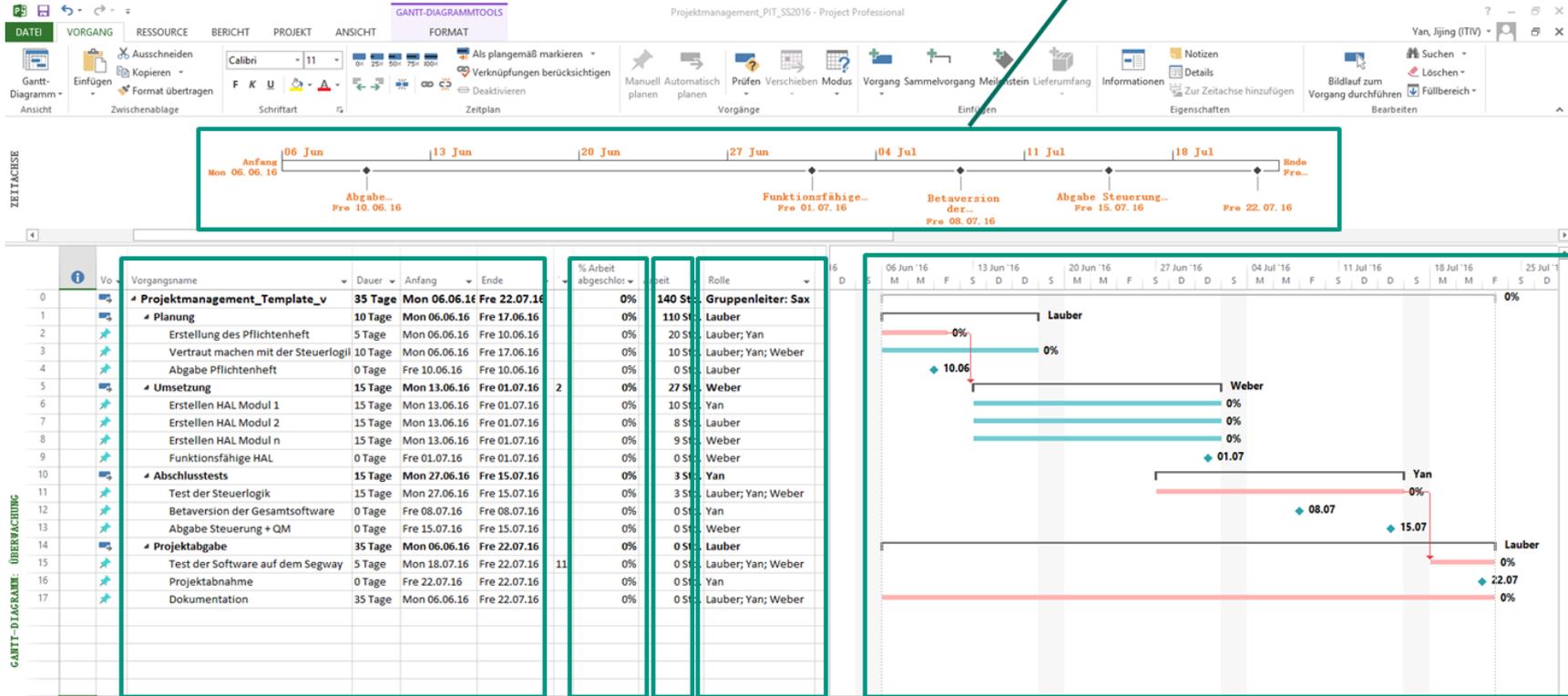
ID	Arbeitspaket	Aufwand (in h)	Dauer (in h) (optional)	Vor- gänger	Nach- folger

Gantt-Chart



Projektmanagement Template für PIT

Microsoft Project 2013



Zeitachse mit 5 Meilensteine

Abgeschlossene Arbeit in % angeben

Zuständige Verantwortlicher

Gantt-Diagramm mit Überwachung

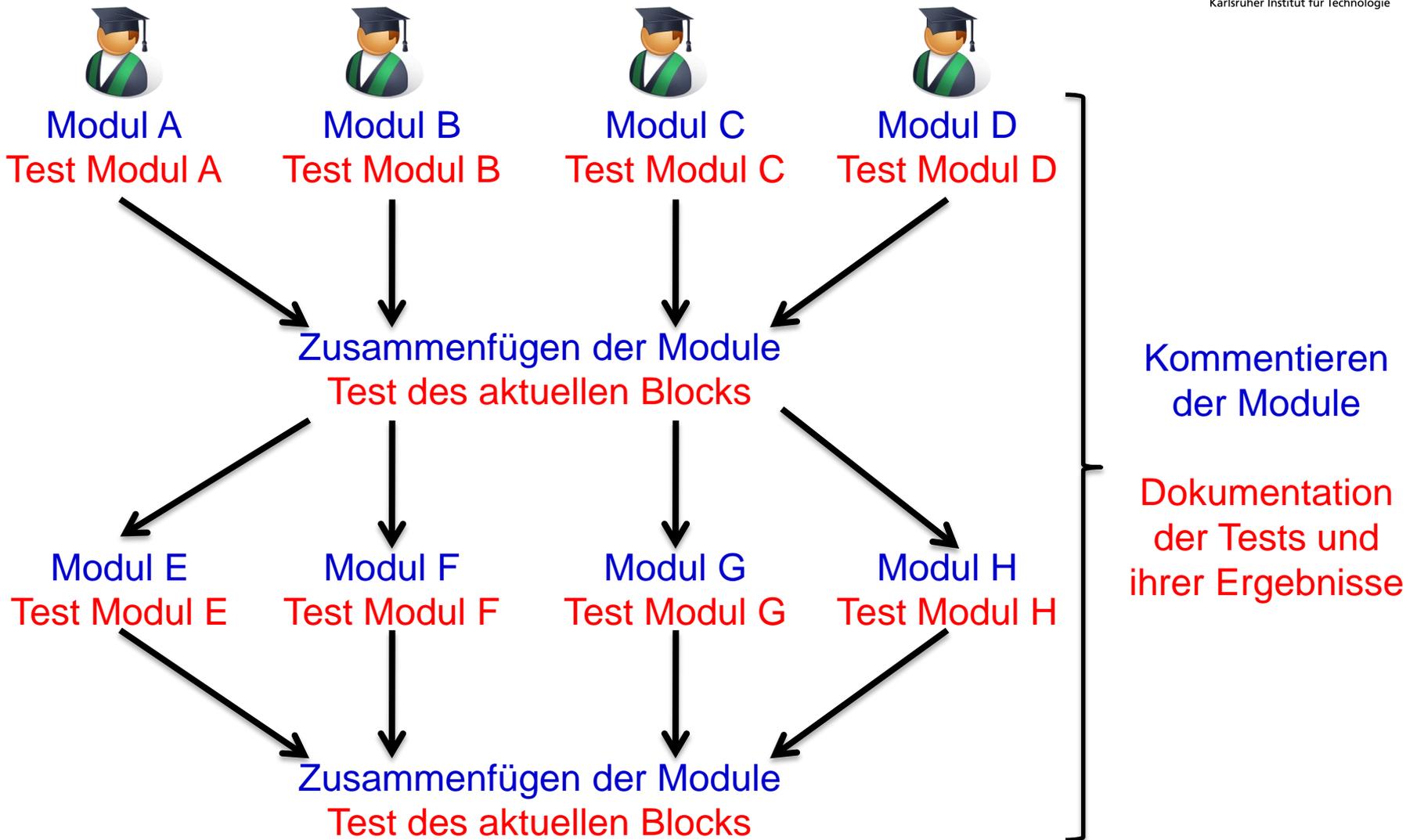
Arbeitspakete

Arbeitsaufwand

2.-4. Termin: Umsetzung

- Ziel: Implementierung und Test des Hardware Abstraction Layers anhand der Spezifikation und Planung
- Gegenseitige Hilfestellung & Hilfestellung durch den Tutor
- Implementierung & Testen der Programme
 - 1. Implementierung von einzelnen Modulen
 - 2. Test der einzelnen Modulen
 - 3. Zusammenfügen der Module
 - 4. Test des zusammengefügteten Programms
 - 5. Beginn wieder bei Punkt 1 bis zur Implementierung des vollständigen Programms
- Liefergegenstände:
 - Hardware Abstraction Layers inkl. Kommentierung und Testdokumentation
 - Einhaltung der Programmierrichtlinien

Bsp.: Implementierung und Testen



5. & 6. Termin: Abschlusstest

- Ziel: Verifikation und Validation des Gesamtprogramms
- Testen der korrekten Lauffähigkeit des Programms
- Einhaltung der Spezifikationen
- Abschlusstest anhand einzelner Tests auf dem TivSeg
- Liefergegenstände:
 - Gesamtprogramm inkl. Kommentierung und Testdokumentation
 - Einhaltung der Programmierrichtlinien
 - Programm muss im Simulator lauffähig sein
 - Projektdokumentation
 - Siehe nächste Folien

7. Termin: Dokumentation und Vorführung



Semester:

WS 2015/16

Projektgruppe:

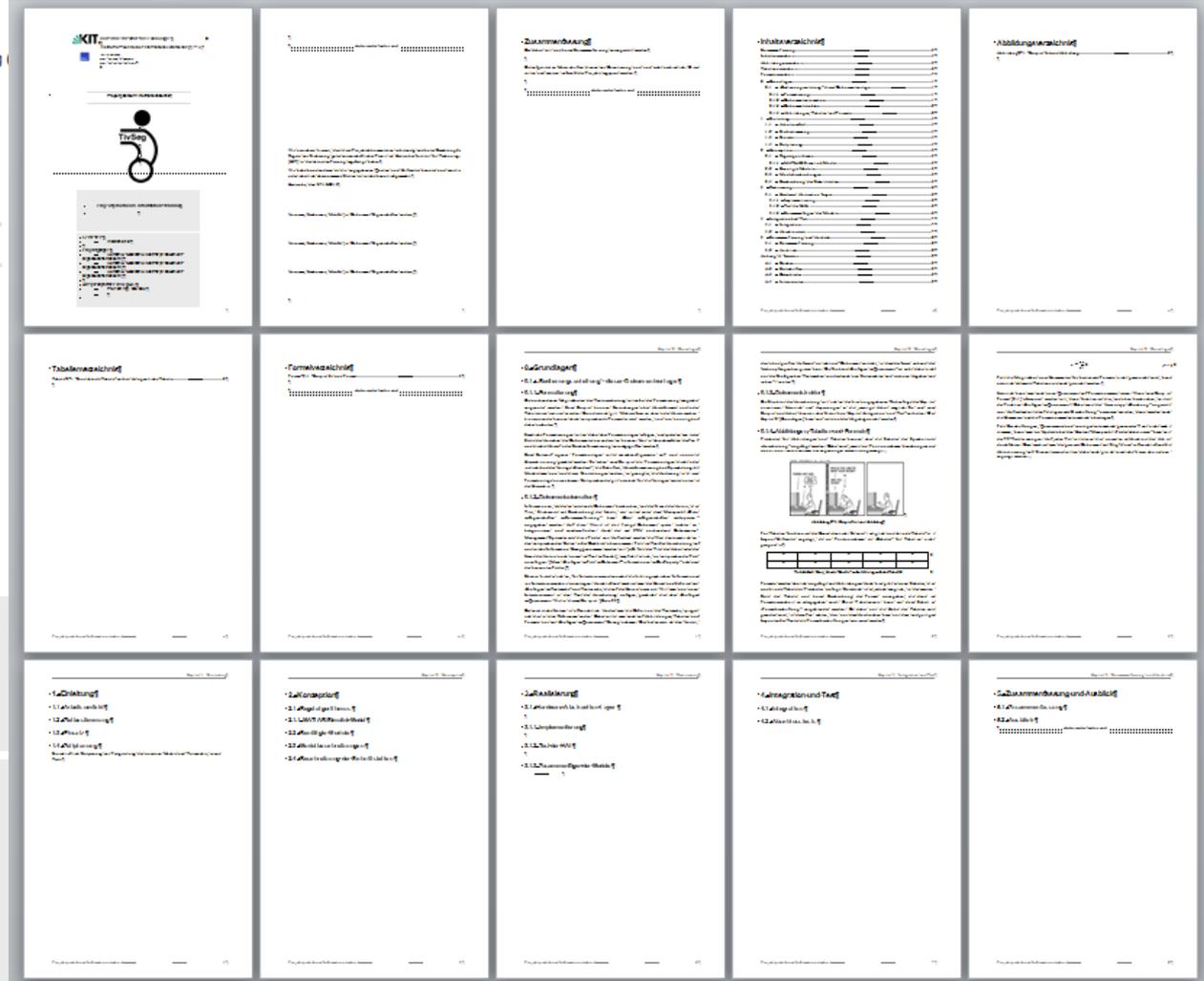
Vorname, Nachname, MatrNr (in Dokument-Eigenschaften ändern)

Vorname, Nachname, MatrNr (in Dokument-Eigenschaften ändern)

Vorname, Nachname, MatrNr (in Dokument-Eigenschaften ändern)

Ansprechpartner Tiv Seg AG:

Prof. Dr.-Ing. Eric Sax



Inhalt der Projektdokumentation

- Aufgabenstellung (SOLL-Konzept)
- Anforderungen
- Projektziele (SOLL-IST-VERGLEICH)
- Projektstrukturplan mit Zeitplanung / Meilenstein (Projektmanagement)
- Beschreibung verwendeter nicht einfacher Funktionen und Algorithmen
- Beschreibung der Tests und ihrer Ergebnisse
- Ausblick / Folgeaktivitäten
- Fazit
- Anlagen / Anhang:
 - Pflichtenheft
 - Kundeneinweisung
 - Abnahme- und Testprotokoll

7. Termin: Vorführung und Probefahrt

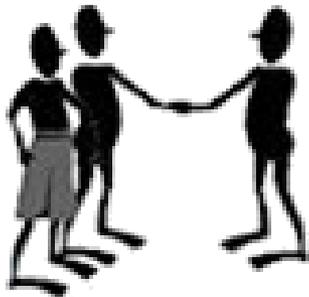
- Die Teilnahme an der Probefahrt ist freiwillig



Stimmungsphasen

Forming

Team establishes ground rules. Formalities are preserved and members are treated as strangers



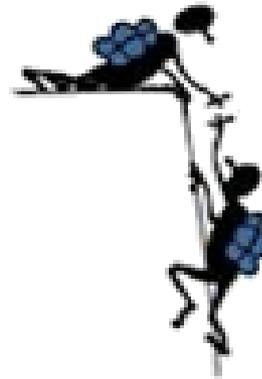
Storming

Members start to communicate their feelings but still view themselves as individuals rather than part of the team. They resist control by group leaders and show hostility



Norming

People feel part of the team and realize that they can achieve work if they accept other viewpoints



Performing

The team works in an open and trusting atmosphere where flexibility is the key and hierarchy is of little importance



DHD

- Bilddaten Vorverarbeitung
- VHDL
- Kommunikation mit PSE - μ C

PSE

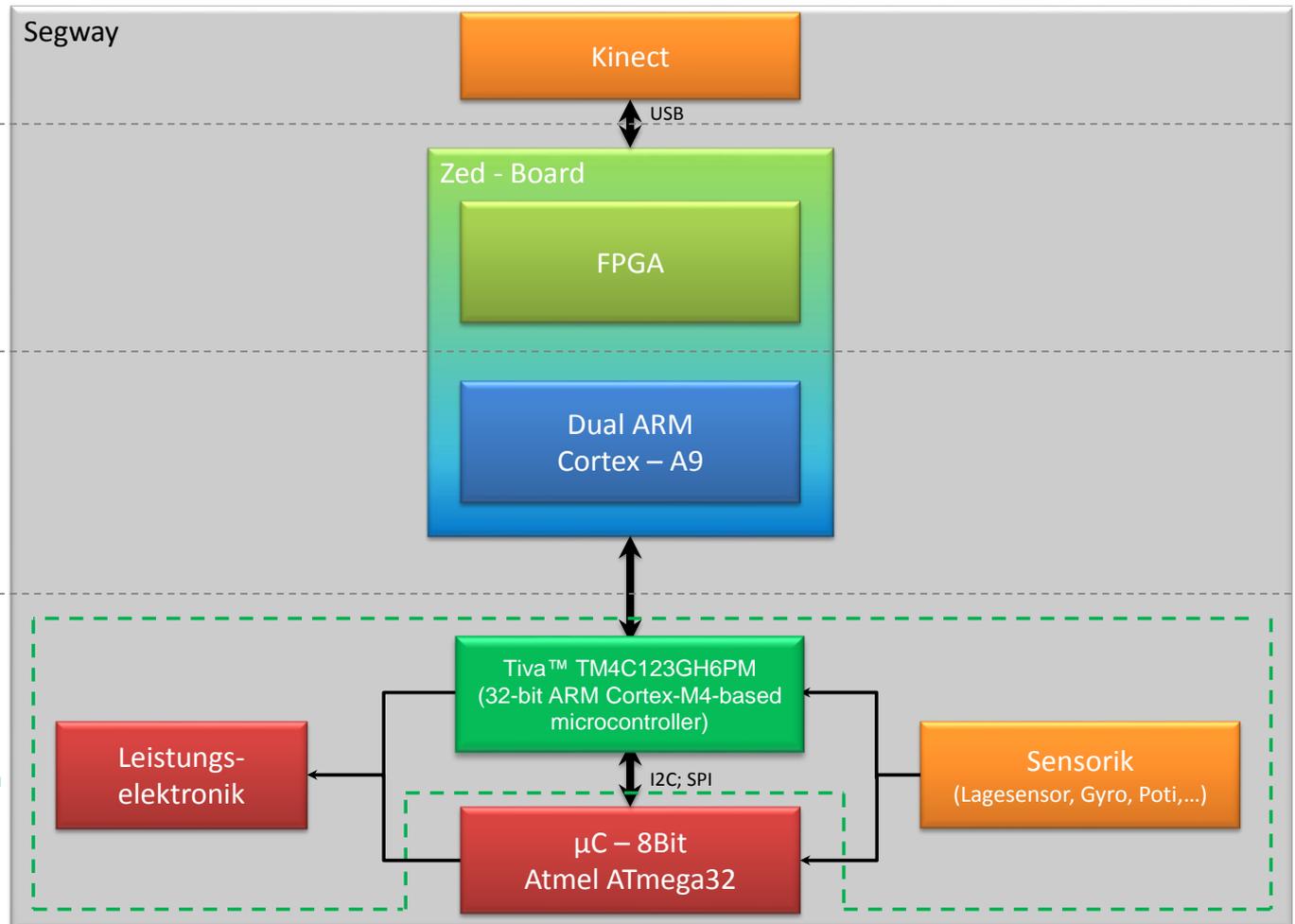
- Verarbeitung Bilddaten
- Regelung über Bilddaten
- Regelung über Sensorik
- Steuerbefehle für LSD - μ C

LSD

- Schaltungsdesign
- Platinenlayout

PIT

- Hardwarenahes Programmieren
- Basics (Tool-Chain, C++)
- „Algorithmen“
- Strukturiertes Programmieren
- Abgestimmt mit IT



2. Einführungsveranstaltung

Inhalt



1

• Tiva C Series TM4C123G LaunchPad Evaluation Board

2

• UART, GPIO, Timer, PWM, ADC, Watchdog im eingebetteten System

3

• Einführung in Register

4

• Umgang mit Datenblättern

5

• Blinky Beispiel

15.05.2018,
17:30- 19:00
a. F. Hörsaal

3

09.05.2017

KIT – Universität des Landes Baden-Württemberg und
nationales Forschungszentrum in der Helmholtz-Gemeinschaft

© TivSeg AG

Tutorium (Selbststudium)

- Kapitel 6 im Lastenheft: Einstieg in die hardwarenahe Programmierung
 - Erstellen eines CCS-Projekts
 - Umgang mit Datenblättern
 - Flashen und Debuggen
 - Beispielprogramm
 - LED Blinken Beispiel zum Erweitern: GPIO, ADC, PWM werden verwendet
- Lösungen für die Aufgaben findet man später in ILIAS



Support

- Tutoren
- Betreuer
 - M. Sc. Jijing Yan, yan@kit.edu
 - Prof. Dr.-Ing. E. Sax, Eric.Sax@kit.edu



...der Geburtsort der Informatik