

Internet of Things – Technical point of view (W3WI_SE407)

| Formale Angaben zum Modul | | |
|---------------------------|----------------------|------------|
| Studiengang | Studienrichtung | Vertiefung |
| - | Software Engineering | - |

| Modulbezeichnung | Sprache | Nummer | Version | Modulverantwortlicher |
|--|------------------|------------|---------|--|
| Internet of Things – Technical point of view | Deutsch/Englisch | W3WI_SE407 | 1 | Reichwald, Prof. Dr. Julian; Mannheim Holey, Prof. Dr. Thomas; Mannheim |

| Verortung des Moduls im Studienverlauf | | | |
|--|-----------------------------------|---------------------------|------------|
| Semester | Voraussetzungen für die Teilnahme | Modulart | Moduldauer |
| 2. Stj. | | Studienrichtungswahlmodul | 2 |

| Eingesetzte Lehr- und Prüfungsformen | |
|--------------------------------------|--|
| Lehrformen | Vorlesung, Übung, Laborübung, Case Study |
| Lehrmethoden | - |

| Prüfungsleistung | Prüfungsumfang (in min) |
|---|-------------------------|
| Kombinierte Modulprüfung | - |
| Bestandteile Kombinierte Prüfungsleistung | |
| Klausur, Assignment | |

| Workload und ECTS | | | |
|---------------------------|--------------------------|----------------------------|-------------|
| Workload insgesamt (in h) | davon Präsenzzeit (in h) | davon Selbststudium (in h) | ECTS-Punkte |
| 150,0 | 55,0 | 95,0 | 5 |

| Qualifikationsziele und Kompetenzen | |
|---|---|
| Fachkompetenz | Die Studierenden kennen den technischen Aufbau typischer Devices/Embedded Systems im Kontext von Internet of Things (IoT). Sie sind in der Lage entsprechende Devices für einen gegebenen Einsatzzweck auszuwählen und zu programmieren |
| Methodenkompetenz | Die Studierenden sind in der Lage bei der Programmierung von IoT-Geräten systematisch und methodisch vorzugehen. |
| Personale und Soziale Kompetenz | Die Studierenden verstehen die Herausforderungen des IoT für Unternehmen, Politik und Gesellschaft und sind in der Lage, diese kompetent zu diskutieren. |
| Übergreifende Handlungskompetenz | Die Studierenden können reale betriebliche Problemstellungen im Kontext von IoT analysieren, Konzepte entwerfen und IoT-fähige Geräteprogrammieren und im Unternehmenskontext integrieren. |

Lerneinheiten und Inhalte

| Lehr- und Lerneinheiten | Präsenz | Selbststudium |
|---|-------------|---------------|
| Hardwarenahe Programmierung und Embedded Systems I | 28,0 | 47,0 |
| Ausgewählte Themen aus dem Bereich Hardwarenahe Programmierung wie z.B.: | | |
| <ul style="list-style-type: none">- Konzept Systemaufbau und Komponenten: CPU, Hauptspeicher, I/O: Diskussion Anbindung externer Geräte (Grafik, Tastatur, Festplatten, DVD, ...)- Halbleiterspeicher- Wahlfreie Speicher: Aufbau, Funktion, Adressdekodierung, interne Matrixorganisation- RAM: statisch, dynamisch, aktuelle Entwicklungen- ROM: Maske, Fuse, EPROM, EEPROM, FEPRM, aktuelle Entwicklungen- Systemaufbau- Aufteilung des Adressierungsraumes- Entwerfen von Speicherschemata und der zugehörigen Adress-Dekodierlogik- Vitale System-Komponenten: Stromversorgung, Rücksetzlogik, Systemtakt, Chipsatz- Schaltkreise: Interrupt- und DMA-Controller, Zeitgeber- und Uhrenbausteine- Schnittstellen: Parallel und seriell, Standards (RS232, USB, ...)- Performancekonzepte- Befehlssatz und Maschinenprogrammierung- Programmiermodell: Befehlssatz und Adressierungsarten- Umsetzung von Kontrollstrukturen, Auswertung von Ergebnisflags- Unterprogrammstruktur mit Hilfe des Hardwarestapels: Mechanismen, Aufruf- Konzept und Umsetzung (HW- und SW-Interrupts): Diskussion von HW- und SW-Mechanismen und Automatismen, IR-Vektortabelle; Spezialfall: Bootvorgang- Diskussion User- und Supervisor-Modus von Prozessoren | | |
| Hardwarenahe Programmierung und Embedded Systems II | 27,0 | 48,0 |
| Ausgewählte Themen aus dem Bereich Embedded Systeme wie z.B.: | | |
| <ul style="list-style-type: none">- Entwurf von Embedded Systemen- Beschreibung des Systemkontexts und Systemzwecks- Dienstspezifikationen- Schnittstellenspezifikation- Grundlagen der Firmwareentwicklung- Modellierung (z.B. UML für Embedded)- Umsetzung- Benutzung von Peripherieeinheiten- Teststrategien- Einführung Hardware-Software-Co-Design- Vernetzung von Embedded Systemen- Konzepte (User Interface, Speicherverwaltung, Ressourcen-limitiertes Computing, HTML5-Ansatz, Native Apps)- Plattformen (z. B. iOS, Android, Blackberry, Windows Phone 7)- Frameworks und Bibliotheken(z. B. Titanium, PhoneGap, Cocoa Touch, com.google.android)- Entwicklungsumgebungen (z. B. Eclipse, Xcode)- Praktische Übungen | | |

Besonderheiten und Voraussetzungen

Besonderheiten

-

Voraussetzungen

-

Literatur

- Biller, S.: Vernetzung von Mikrocontrollern mit dem Ethernet: TCP/IP Kommunikation für Mikrocontroller und Embedded-Systeme", VDM Verlag Dr. Müller
- Conway, J. & Hillegass, A.: iPhone Programming: The Big Nerd Ranch Guide, Addison-Wesley Longman
- Gray, J. S.: Interprocess Communications in UNIX, Prentice Hall
- Hillegass, A.: Cocoa Programming for Mac OS X, Addison-Wesley Longman
- Kochan, S. G.: Programming in Objective-C 2.0, Addison-Wesley Longman
- Petzold, C.: Windows Programmierung mit C#, Microsoft Press
- Quinn, B. & Shute, D.: Windows Sockets Network Programming, Addison-Wesley
- Spiering, M. & Haiges, S.: HTML5-Apps für iPhone und Android, Franzis-Verlag
- Steele, J. & To, N.: The Android Developer's Cookbook: Building Applications with the Android SDK, Addison-Wesley Professional
- Templeman, J. & Olsen, A.: Visual C++ Schritt für Schritt, Microsoft Press