

# Mikroprozessortechnik

**SS24**

# Vorstellung

- ca. 1976: Beginn mit Elektronik (~~Analog~~ -> Digital)
- ca. 1978: Beginn mit Mikroprozessoren
- ca. 1982: Beginn mit Hochsprachen (~~Pascal~~ -> C)
- 1985-1995: Studium & Promotion E-Technik/Informatik (TUM)
- 1995-2000: Siemens HL -> Infineon
- seit WS02: HM
  - Mikroprozessortechnik (MEB4)
  - Industriepraktikum (MEB6)
  - *Embedded Systems II* (MEB7)
  - Digitalelektronik (MFM)
  - Informatik, Digitale Signalverarbeitung, Netzwerktechnik

# Einführung

- **Stellung im Studiengang**

- 3. Semester: Informatik (real: „Ich lerne C“)
- 4. Semester: **Mikroprozessortechnik** (real: „C auf Mikrocontrollern“)

- **Vorlesung**

- Ergänzung/Vertiefung **C für Mikrocontroller**
- Entwurf und Umsetzung eines **Automaten**
- Funktion einiger ausgewählter **Hardwaremodule** eines Mikrocontrollers

- **Praktikum**

- Versuche mit einem Mikrocontrollersystem (Präsenz/mit Kit Online)

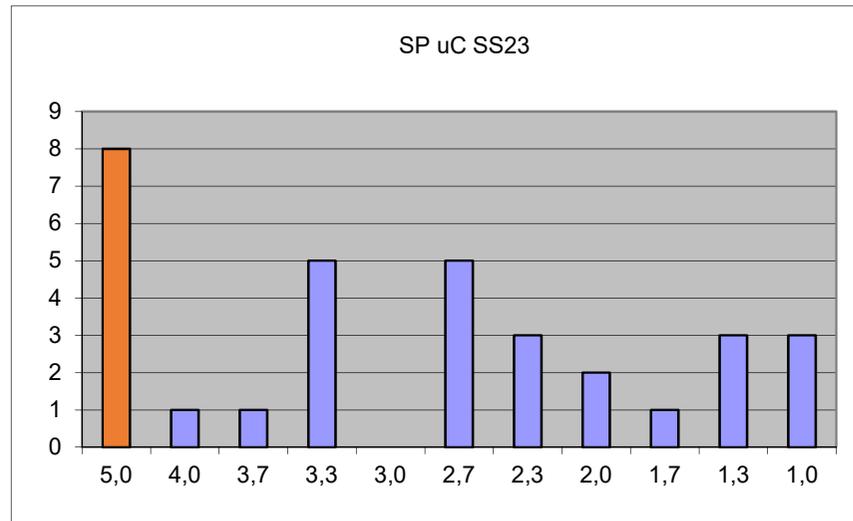
# Organisation

- **Vorlesung** (Planung 19. März)
  - 1x Einführung
  - 3x Vertiefung/Ergänzung C
  - 3x Automat
  - 5x Hardware Mikrocontroller
  
- **Praktikum (optional)**
  - *Ausleihmöglichkeit Experimentierkit (25€ Pfand)*
  - 6 Versuche (je ca. 14 Tage Bearbeitungszeit), Beginn nach Ostern
  - Alleine oder in Zweiergruppen
  - Verbesserung des Verständnisses
  - C-Anteil prinzipiell ohne Kit möglich (z.B. mit Visual Studio)

# Prüfung

- **Vorlesung**

- SP, 60 Minuten
- mit allen Unterlagen



- **Praktikum**

- Bewertung pro Versuch und Gruppe
- 5 Versuche müssen bestanden werden (d.h. Bewertung  $\geq 50\%$ ),
- Endergebnis: Mittelwert der 4 besten Versuchsergebnisse
- Gewichtung: 30%, falls besser als SP, sonst keine Auswirkung

# Mikroprozessor / Mikrocontroller

- **Mikroprozessor (CPU,  $\mu$ P)**
  - „nackter“ Rechenkern (aber dafür i.d.R. sehr leistungsfähig)
  - Desktop-Rechner, Notebooks, typisch Intel/AMD
- **System-On-A-Chip (SoC)**
  - CPU für allgemeine Aufgaben und Betriebssysteme wie Linux
  - Grafik und Kommunikation integriert, aber externer Speicher nötig (Gigabyte)
  - Mobiltelefone, High-End-Konsumgeräte (TV, ...), Raspberry Pi-Klasse
- **Mikrocontroller ( $\mu$ C)**
  - „Pizza con tutti“: von allem etwas integriert (inklusive Speicher)
  - unzählige Varianten, je nach Anwendung passend einen aussuchen
  - weiße Ware, Gerätesteuerungen, Arduino-Klasse

# Unterschiede $\mu P$ -> SoC -> $\mu C$

- In einen  $\mu C$ -System ist (oft) **nicht vorhanden**
  - Betriebssystem (oder stark eingeschränkt)
  - Standardumgebung (Tastatur, Bildschirm)
  - Massenspeicher (Festplatte)
- Programme werden auf einem **anderen System entwickelt** (PC)
- Es gibt deutliche **Beschränkungen** bei
  - Rechenleistung
  - verfügbarem Speicher (Arbeitsspeicher)
  - Energieverbrauch
  - Kosten pro Stück

# Programmentwicklung

- Standardsprache ist (noch) C
- Entwicklungsumgebungen (IDE) herstellerspezifisch
  - sehr häufig kostenfreie und gute IDE vom  $\mu$ C-Hersteller verfügbar
  - ähnlicher Aufbau (Basis Eclipse und gcc)
- Für Basisaufgaben Bibliotheken verfügbar (auch herstellerspezifisch)
- Entwicklung über speziellen Adapter PC  $\rightarrow$   $\mu$ C (Debugger)
- Häufige Verwendung des Automatenmodells
- Häufige Verwendung von Programmunterbrechungen
- Portabilität über Herstellergrenzen
  - hängt von der Aufgabenstellung ab
  - hängt von der eigenen Planung vor(!) Entwicklungsbeginn ab

# C, C, C

- **Die guten Nachrichten**

- Man muss gar nicht so besonders gut C können
- Vieles wird heute über fertige Bibliotheken erledigt

- **Die schlechten Nachrichten**

- nicht immer sind die Kenntnisse (noch) da
  - Bsp.: Wie rufe ich noch mal eine Funktion mit Argumenten auf?
- **Umgang mit Zeigern nötig**,
  - werden beim Aufruf von Bibliotheksfunktionen verwendet, nicht änderbar
- **Aufzählungen** (enum) und **Strukturen** (struct), **neue Typen** (typedef)
  - werden beim Aufruf von Bibliotheksfunktionen verwendet, nicht änderbar
- Ein wenig mehr C:
  - static, volatile, **bedingte Kompilierung**, **stdint-Typen**