Mikroprozessortechnik

2022-10-04

Einführung

- **Stellung im Studiengang**
  - 3. Semester: Informatik (real: „Ich lerne C“)
  - 4. Semester: Mikroprozessortechnik (real: „C auf Mikrocontrollern“)
  - 5. Semester: Embedded Systems I (real: „Design mit Mikrocontrollern“)

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

- **Praktikum**
  - Versuche mit einem Mikrocontrollersystem (auch zuhause möglich)

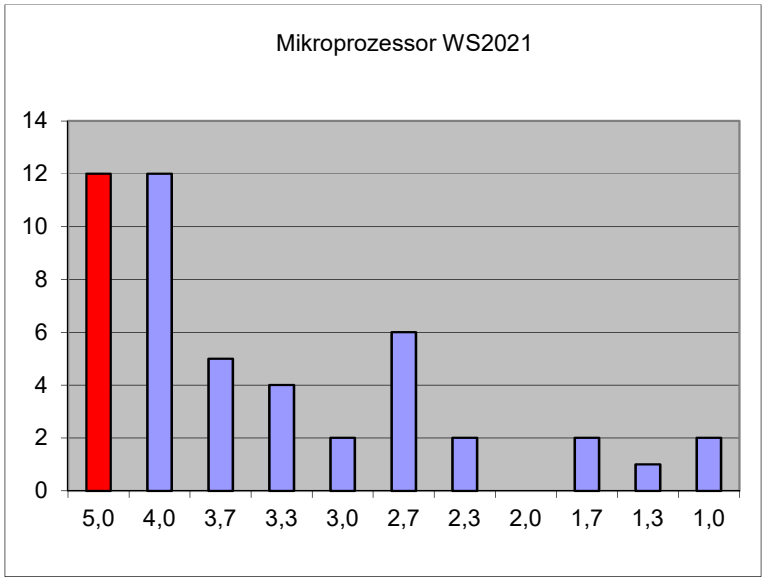
Organisation

- **Vorlesung** (Planung 4. Oktober)
  - 1x Einführung
  - 3x Vertiefung/Ergänzung C
  - 3x Automat
  - 5x Hardware Mikrocontroller

- **Praktikum (freiwillig)**
  - 6 Versuche (je ca. 10 Tage Bearbeitungszeit), Beginn 18. Oktober
  - Zweiergruppen (auch wenn zuhause)
  - Begrenzte Ausleihmöglichkeit Experimentierkit (25€ Pfand)

Prüfung

- **Vorlesung** (Stand 4. Oktober)
  - SP am Ende
  - 60 Minuten Zeit
  - mit allen Unterlagen



- **Praktikum (freiwillig)**
  - Verbesserung des Verständnisses
  - C-Anteil ist ohne Kit möglich (z.B. mit Visual Studio)
  - Verbesserung der Note möglich (dann 30%)
  - Bewertung pro Versuch und Gruppe am Ende der Bearbeitungszeit
  - Mindestteilnahme für Notenverbesserung: 5 Versuche mit ≥ 50% Erfolg

Mikroprozessor / Mikrocontroller

- **Mikroprozessor (CPU, µP)**
  - „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 (µC)**
  - „Pizza con tutti“: von allem etwas integriert (inklusive Speicher)
  - unzählige Varianten, je nach Anwendung passend einen aussuchen
  - weiße Ware, Gerätesteuerungen, Arduino-Klasse

Unterschiede µP -> SoC -> µC

- In einen µC-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 C
- Entwicklungsumgebungen (IDE) herstellerspezifisch
  - sehr häufig kostenfreie und gute IDE vom µC-Hersteller verfügbar
  - ähnlicher Aufbau (Basis Eclipse und gcc)
- Für Basisaufgaben Bibliotheken verfügbar (auch herstellerspezifisch)
- Entwicklung über speziellen Adapter PC -> µC (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

Früher µC (ca. 1976)

Früher µC (ca. 1976)

Device	Internal Memory		RAM Standby
8050AH	4K x 8 ROM	256 x 8 RAM	yes
8049H	2K x 8 ROM	128 x 8 RAM	yes
8048H	1K x 8 ROM	64 x 8 RAM	yes
8040AHL	none	256 x 8 RAM	yes
8039HL	none	128 x 8 RAM	yes
8035HL	none	64 x 8 RAM	yes
8749H	2K x 8 EPROM	128 x 8 RAM	yes
8748H	1K x 8 EPROM	64 x 8 RAM	yes

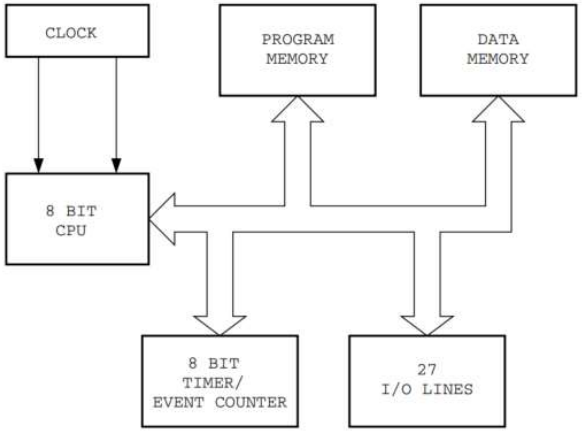


Figure 1.
Block Diagram

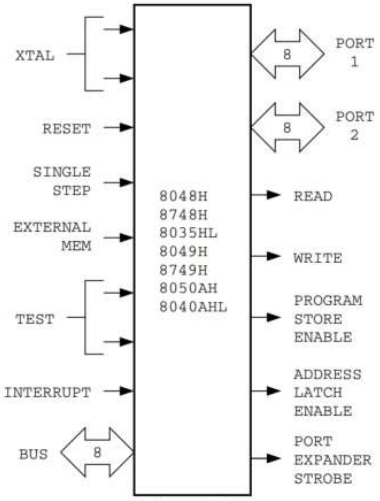


Figure 2.
Logic Symbol

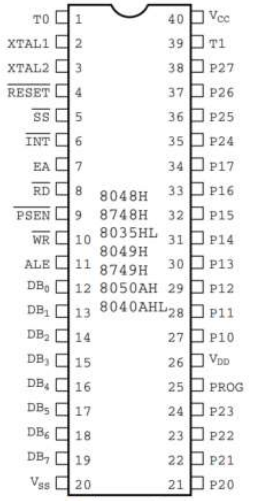


Figure 3.
Pin Configuration