



Programmierung von Retro Spielen auf einer verteilten LED Matrix

IBR (CM) – SEP Kickoff

Robert Hartung, Linda Fliß, Torben Petersen, 2019-04-10

[Allgemeines](#) [Ablauf und Anforderungen](#) [Bewertung](#) [Aufgabenstellung](#)

Allgemeines

- Terminkalender auf der Webseite¹
- SEP besteht aus
 - Dokumenten
 - Veranstaltungen
 - Programmierung
- Teilnahme an allen Veranstaltungen ist Pflicht!
- Programmieren ist Pflicht!
- Dokumente sind eine Gruppenleistung!
 - Editieren von XML-Dateien ist nicht ausreichend

¹<https://www.ibr.cs.tu-bs.de/courses/ss19/sep-cm/index.html>

[Allgemeines](#) [Ablauf und Anforderungen](#) [Bewertung](#) [Aufgabenstellung](#)

Gruppen

- 2 Gruppen: IBR_CM0, IBR_CM1
- Jeder Teilnehmer muss zu allen Phasen beitragen
- Organisation innerhalb der Gruppe ist frei
 - Eigenständige Aufteilung bei Dokumente (Kapitel, nicht Bild/Text)
 - Sinnvolle Aufteilung der Programmierung (Komponenten)
- Alle müssen auf Termine achten ⇒ Teamleistung!

Vorgehensmodell

Vereinfachtes V-Modell inkl. frühem Prototyp

- Angebot (Aufgabenverständnis)
- Anforderungsanalyse mit Spezifizierungen von Systemtests
Pflichtenheft/Abnahmetestspezifikation
- Prototyp – Zwischenpräsentation
- Fachentwurf *Systemspezifikation 1*
- Technischer Entwurf *Systemspezifikation 2*
- Testen (Systemtests, Integrationstests, Unit-Tests)

Dokumente

Abgabe

- Für jede Phase eine Dokumentvorlage in \LaTeX
- Vorlagen dürfen (nur Begründet!) angepasst werden
- Inhalte/Diagramme sind sinnvoll zu befüllen
- Termine für Abgaben **unbedingt** einhalten
- Abgaben beim ISF im Redmine unter Dokumente
 - **Abgaben können auch iterativ erfolgen!**
 - Im Redmine: README.pdf lesen
- **Vorabgaben bei Linda sind Pflicht (eine Woche vorher)**

Erfahrungen

- Keine Beispieltexthe in Dokumente! Alles ändern!
- Zu jeder Überschrift einen Absatz schreiben
- Bearbeitertabelle sinnvoll nutzen

Arbeitsumgebung

Redmine

- Projektmanagement/Redmine/SVN/Ticketsystem
- Vom ISF gestellt: <https://sep.isf.cs.tu-bs.de/>

SVN

- SVN-Clients: z.B. TortoiseSVN, RabbitSVN
- Nutzung sinnvoll und Pflicht
- Sinnvolle Commit-Logs erstellen
- https://sep.isf.cs.tu-bs.de/svn/19-ibr-cm_X

Mailingliste

Gemeinsame Mailingliste (IBR)

- SEPCm@ibr.cs.tu-bs.de

Spezifische Mailingliste

- SEPCmX@ibr.cs.tu-bs.de

stehen alle auf der Liste?

Kontaktpersonen

HiWis

- **Linda Fliß**
Erreichbar über Mailingliste oder betreute Übungszeiten
- Torben Petersen (Hardware)
- Puhang Ding (Software auf ATmega)

Betreuer

- Robert Hartung
 - Erreichbar per E-Mail an die ML (sonst hartung@ibr.cs.tu-bs.de), falls Fragestunde / regelmäßige Treffen nicht ausreichen

Betreute Übung

SEP-Block

- Do. 16:45 - 18:15
- Kann intern auch geändert werden

Terminfindung in den jeweiligen Gruppen

- Besprechen wir gleich in großer Gruppe

Bewertung

- ISF korrigiert Dokumente und legt Noten fest
- Rückgabe der Dokumente nach ca. 10 Tagen
- Wir legen individuelle Noten fest
- Das SEP **ist benotet**, das ISF meldet die Noten dem jeweiligen Prüfungsamt
- Jeder Teilnehmer muss sich bei der Programmierung mit einbringen
- Benotungsrichtlinien des ISF (README in den Vorlagen²)

²<https://sep.isf.cs.tu-bs.de/redmine/attachments/5139/vorlagen.zip>

Bewertung

- Realisierung des Funktionsumfangs
- Einhalten der Deadlines
- Arbeit als Phasenverantwortlicher
- Teilnahme an den Präsenzveranstaltungen
- Aktive Mitarbeit in der Gruppe
- Code Review

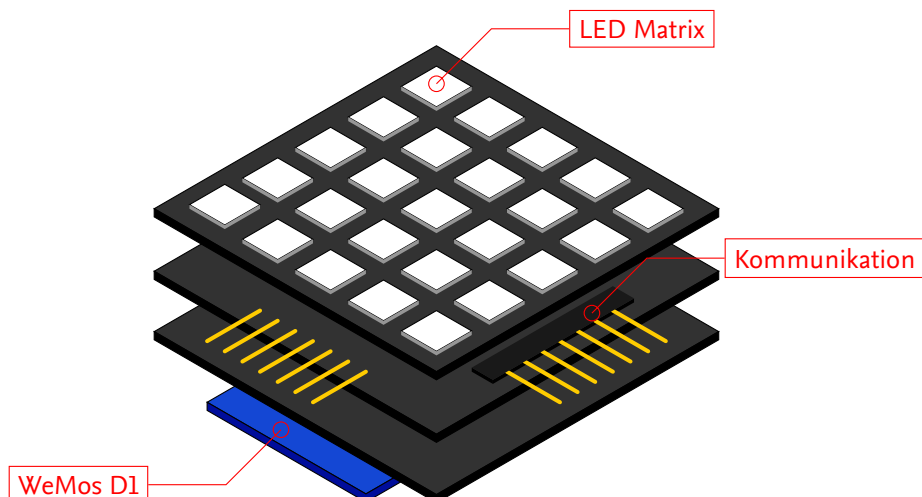
Eure Aufgabe

Idee

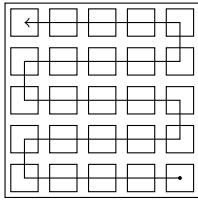
- Inspiration: LED Tisch mit WS2812B LEDs
- Institut für Betriebssysteme und **Rechnerverbund**
 - Statt einer Steuerungskomponente mehrere, verteilte Systeme
- ⇒ Verschiedene Herausforderungen
 - Synchronisation
 - Verteilte Logik
 - Topologie



Systemaufbau – Modul

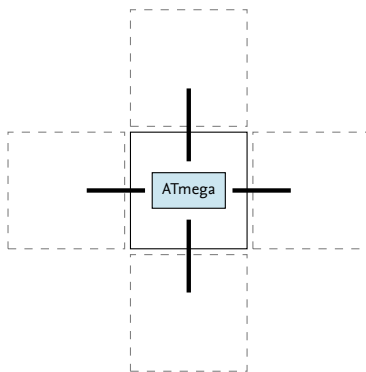


Systemaufbau – LED Matrix



- 25x WS1820B RGB LED
- Ansteuerung über WEMOS D1 (Bibliothek vorhanden)
- Herausforderungen:
 - Reihenfolge der LEDs

Systemaufbau – Kommunikation



- ATmega4809 mit 4 UARTs
- SPI Slave mit eigenem Kommunikationsprotokoll (vorgegeben)
- Herausforderungen:
 - Topologie unbekannt (wer sind meine Nachbarn?)
 - Zeitsynchronisation
 - Nutzereingaben (Bedienung)

Systemaufbau – WEMOS D1

Aufgaben

- Ansteuerung der LEDs
- Kommunikation via SPI mit ATmega und Nachbarknoten
- Darstellung über ein Modul hinweg
- Verarbeiten von Nutzereingaben

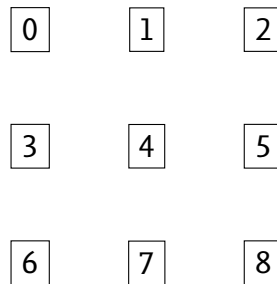
ESP8266

- WLAN Chip
- Mesh oder Access Point für Kommunikation denkbar

Kommunikation – Prinzip

Nachbarerkennung

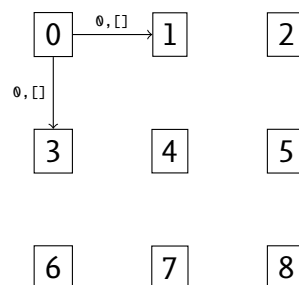
- WEMOS D1 besitzt eindeutige ID
- Strategien zum Aufbau einer Nachbarliste:
 - Beispiel: Verteilen der eigenen und aller bekannten Nachbarn an die Nachbarn



Kommunikation – Prinzip

Nachbarerkennung

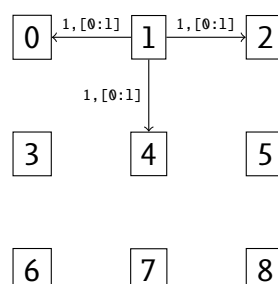
- WEMOS D1 besitzt eindeutige ID
- Strategien zum Aufbau einer Nachbarliste:
 - Beispiel: Verteilen der eigenen und aller bekannten Nachbarn an die Nachbarn



Kommunikation – Prinzip

Nachbarerkennung

- WEMOS D1 besitzt eindeutige ID
- Strategien zum Aufbau einer Nachbarliste:
 - Beispiel: Verteilen der eigenen und aller bekannten Nachbarn an die Nachbarn



Eure Aufgabe(n)

- Kenntniss über Topologie erlangen (SPI, Eigenes Protokoll zwischen WEMOS)
- Ansteuerung der LEDs
- Synchronisation zwischen Modulen
- Eingabemodule (Buttons, Joystick, ...) ⇒ TBD
- Eigentliche Anwendung:
 - Spiele (Tetris, Snake)
 - Musik Visualisierung
 - Text³
- Architektur: Verteilt (besser) oder Server/Client (notfalls)

³<https://www.dafont.com/5x5.font>



Vielen Dank

- Fragen?
- Nächster Schritte:
 - Terminfindung betreute Übungszeit
 - Gruppen für sich: In das Thema einarbeiten
 - Angebot erstellen
 - Was genau wollt ihr programmieren?
 ⇒ Mindestens zwei verschiedene Anwendungen!
 - Einrichten der jeweiligen Arbeitsumgebung
 - SVN/Git, Toolchain (Arduino IDE), ...

