



BEUTH HOCHSCHULE
FÜR TECHNIK
BERLIN
University of Applied Sciences

Dynamische Flüssigkeitssimulation auf einem RGB-Würfel

von

Kayoko Abe, Oleksandra Baga, Heiko Radde

EDV.Nr.:826058, 849852, 887027

Dokumentation des Projektes im Modul
‘Fortgeschrittene ARM Programmierung’
von Herrn Prof. Dr. Görlich

Tag der Abgabe 1. März 2019

Inhaltsverzeichnis

1	Einführung	1
2	Implementierung	3
2.1	Entwicklungsumgebung	3
2.2	Flüssigkeitssimulation	3
2.3	Gyro	3
2.4	Display	3
2.4.1	Hardware	3
2.4.2	Software	4
3	Ergebnis	7

Abbildungsverzeichnis

2.1	Aufbau von sechs 32×32 RGB Matrix LED-Panels	4
2.2	Pinbelegung	5

Quelltextverzeichnis

Kapitel 1

Einführung

Im Rahmen des Moduls ‘Fortgeschrittene ARM Programmierung’ wird unser Projekt *Dynamische Flüssigkeitssimulation auf einem RGB-Würfel* auf dem Keil Evaluation Board MCB2388 realisiert. Der Board installiert einen NXP LPC2388 Microcontroller und besitzt verschiedene Schnittstellen wie XXX, YYY, ZZZ.

Kapitel 2

Implementierung

2.1 Entwicklungsumgebung

Zur Entwicklung der Softwarekomponente soll erst Keil[®] Microcontroller Development Kit (MDK) auf einem Rechner installiert werden. Das Kit ist besonders gut geeignet für Arm[®]-basierte Microcontroller und beinhaltet alle notwendigen Komponente für Embedded Applikationen. Die Edition MDK-Lite ist kostenlos jedoch mit der Beschränkung von 32KBytes Codegröße verfügbar. Zusätzlich muss Legacy Package ARMv7 (NXP LPC 2388) installiert werden. Anschließend soll die Integrierte Entwicklungsumgebung Keil μ version 5 erfolgreich installiert sein.

Der Board kann durch einen USB-Anschluss vom Rechner eingeschaltet werden. Der ULINK-ME kann ebenfalls durch einen USB-Anschluss zum Laden des Programms und zum Debuggen verwendet werden.

2.2 Flüssigkeitssimulation

2.3 Gyro

2.4 Display

Die Komponente Display empfängt Daten der Panelinformationen von der Komponente Flüssigkeitssimulation und beleuchtet die Panels entsprechend dieser Daten.

2.4.1 Hardware

Display des RGB-Würfels wird aus sechs 32×32 RGB Matrix LED-Panels aufgebaut. Ein Panel benötigt den Strom von 2A. Die Panels werden anhand einer Stromversorgung in Abbildung ?? geschaltet. Ein Panel wird über 16 Kabel direkt mit dem Board verbunden und die restlichen fünf Panels werden nebeneinander verbunden (sogenannte Daisy-Chain). Abbildung 2.1 stellt den gesamten Aufbau der Komponente Display dar.

Pinbelegung

Ein Panel kann durch freie GPIO-Pins mit dem Board verbunden werden. Abbildung 2.2 zeigt diese Pinbelegung. Es ist dabei zu beachten, die bereits vom Hersteller belegten Pins

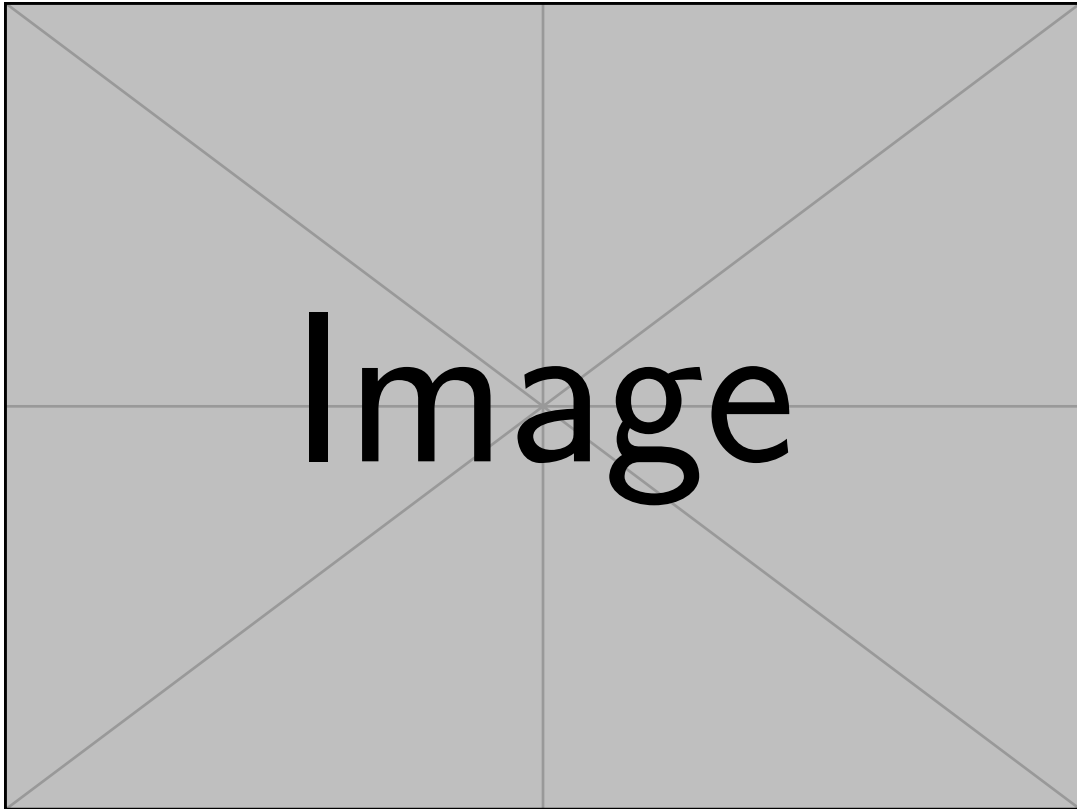


Abbildung 2.1: Aufbau von sechs 32×32 RGB Matrix LED-Panels

zu vermeiden, um den Board erfolgreich in Betrieb zu nehmen.

2.4.2 Software

Initialisierung

Die belegten Pins und die relevanten Ports 2 und 3 müssen wie Code ?? initialisiert werden, um diese als GPIO/Output-Pins verwenden zu können.

Mechanismus und Konfiguration einer Anzeige

Eine Anzeige hat 32 LEDs in einer Reihe beziehungsweise $32 \times 32 = 1024$ LEDs. Diese werden in 16 Teile abgeschnitten. Erster Teil ist für die 1.- und 16. Zeile, zweiter Teil ist für die 2.- und 17. Zeile bis hin zum sechzehnten Teil für die 16.- und 32. Zeile.

<Schritte>

1. Setzen alle Pins auf Low (anfang)
2. R1, G1, B1, R2, G2, B2 Pins auf High/Low setzen
3. CLK Pin auf High setzen
4. CLK Pin auf Low setzen
5. Schritte 2-4 für alle Spalten wiederholen

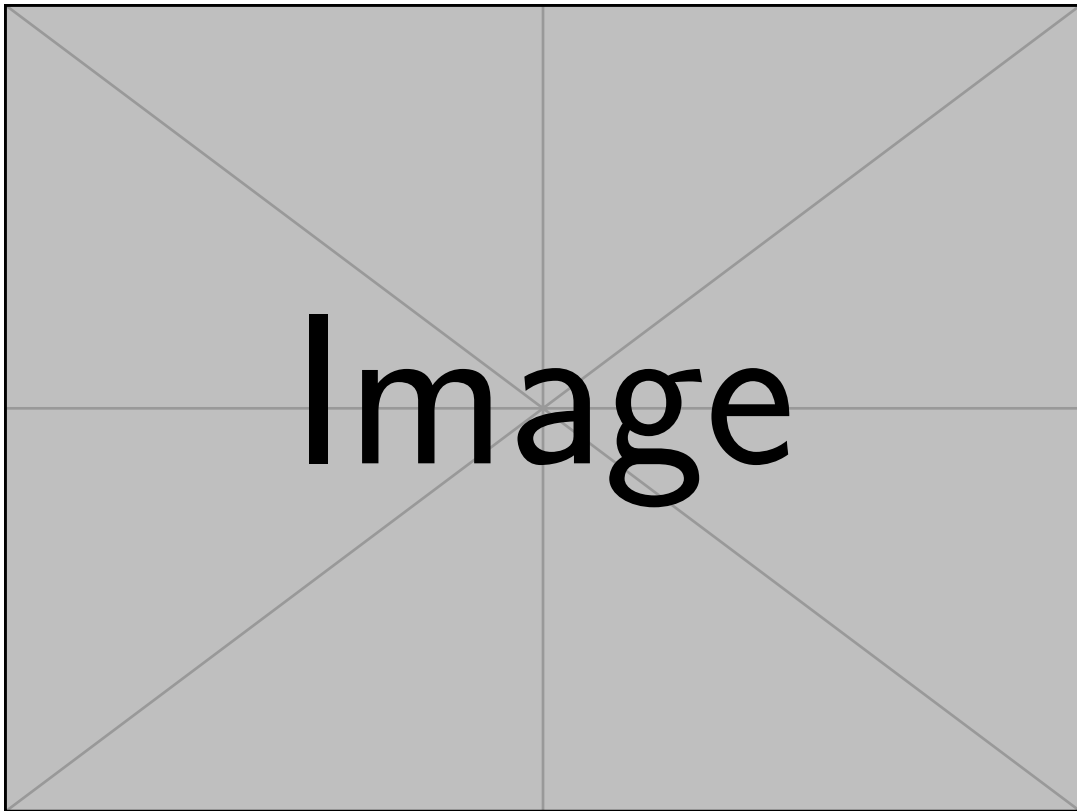


Abbildung 2.2: Pinbelegung

6. Mit Adresspins ABCD eine gezielte Zeile auswählen
7. OE Pin auf High setzen
8. LAT Pin auf High setzen (erleuchtet LED)
9. LAT Pin auf Low setzen
10. OE Pin auf Low setzen

Die Information eines Pixels wird per Takt mittels des Schiebregisters verschoben. Werden mehrere Panels verkettet, wird die Information zum weiteren Panel verschoben. Das heißt, die verketteten Panels können als ein Panel mit den breiteren Spalten betrachtet werden. Jeder Zyklus leuchtet LEDs und schaltet diese gleich aus. Die Unterfunktion `refresh` muss daher kontinuierlich aufgerufen werden.

Kapitel 3

Ergebnis