

# Technische Informatik 3 – Embedded Systems

## Kapitel 3: Dateibearbeitung

Prof. Dr. Benjamin Kormann

Fakultät für Elektro- und Informationstechnik

24.04.2023



# Grundlegende Funktionen zur Dateibearbeitung

## Allgemeine Informationen

- Prinzipiell muss eine Datei vor dem Zugriff auf sie geöffnet und hinterher wieder geschlossen werden
- Dazu sind Funktionen, Datentypen etc. aus der Standardbibliothek `stdio.h` erforderlich

## Übersicht grundlegender Funktionen

Funktion	Beschreibung
<code>fopen()</code>	Öffnen einer Datei
<code>fclose()</code>	Schließen einer Datei
<code>fseek()</code>	Verändern der Bearbeitungsposition
<code>ftell()</code>	Ermitteln der aktuellen Bearbeitungsposition
<code>feof()</code>	Überprüfung auf Dateiende
<code>fflush()</code>	Verarbeiten (leeren) des Dateipuffers

# Grundlegende Funktionen zur Dateibearbeitung

## Datei öffnen

### Syntax

```
FILE *fopen(const char *path, const char *mode);
```

"t" für Textdatei  
"b" für Binärdatei

- path: Pfad zur Datei, die geöffnet werden soll
- mode: Modus mit dem die Datei geöffnet werden soll

### Unterstützte Modus

Modus	Beschreibung
"r"	Lesen (Datei muss vorhanden sein)
"w"	Schreiben (vorhandene Datei wird überschrieben)
"a"	Anhängen am Dateiende (vorhandene Datei wird nicht überschrieben)
"r+"	Lesen und Schreiben (Datei muss vorhanden sein)
"w+"	Schreiben und Lesen (vorhandene Datei wird überschrieben)
"a+"	Anhängen am Dateiende und Lesen an beliebiger Position

# Grundlegende Funktionen zur Dateibearbeitung

## Funktionen für Text- und Binärdateien

### Übersicht von Funktionen zur Bearbeitung von Dateien

Funktion	Beschreibung
<code>fgetc()</code>	Zeichenweises Lesen
<code>fputc()</code>	Zeichenweises Schreiben
<code>fscanf()</code>	Formatiertes Lesen
<code>fprintf()</code>	Formatiertes Schreiben
<code>fgets()</code>	Zeilenweises Lesen
<code>fputs()</code>	Stringweises Schreiben
<code>fread()</code>	Binäres Lesen von Dateien (byteweise)
<code>fwrite()</code>	Binäres Schreiben von Dateien (byteweise)

```
int fputc(int c, FILE *f)
```

```
int fprintf(FILE *f, const char *format, ...) wie printf()
```

```
int fscanf(FILE *f, const char *format, ...) wie scanf()
```

# Grundlegende Funktionen zur Dateibearbeitung

## Beispielprogramm für Textdatei

Überprüfung, ob das Öffnen  
der Datei geklappt hat

```
#include <stdio.h>

int main()
{
    int c;
    FILE* file;
    file = fopen("hallo.txt", "r");
    if (file == NULL)
    {
        printf("Konnte Datei nicht finden bzw. öffnen!\n");
        return -1;
    }

    while ((c = fgetc(file)) != EOF)
    {
        fputc(c, stdout);
    }

    fclose(file);

    return 0;
}
```

Zeichenweise bis EOF lesen  
und nach stdout schreiben

# Grundlegende Funktionen zur Dateibearbeitung

## Beispielprogramm für Binärdatei

Überprüfung, ob das Öffnen  
der Datei geklappt hat

```
#include <stdio.h>

int main()
{
    int iData[8];
    double dData[16];

    FILE* file;
    file = fopen("hallo.bin", "rb");
    if (file == NULL)
    {
        printf("Konnte Datei nicht finden bzw. öffnen!\n");
        return -1;
    }

    fread(iData, sizeof(int), 8, file);

    fread(dData, sizeof(double), 16, file);

    fclose(file);

    return 0;
}
```

Größe eines Elements

Anzahl der Elemente

# Zusammenfassung

## Grundlegende Funktionen für Text- und Binärdateien

- Öffnen, Schließen, Modus, Lesezeiger
- Optionen für Modus zur Dateibearbeitung

## Funktionen zur Bearbeitung von Dateien

- Textdateien: `fgetc`, `fputc`, `fscanf`, `fprintf`, `fgets`, `fputs`
- Binärdateien: `fread`, `fwrite`