# Tutorium zu Computer-Engineering im SS19 Termin 4

Jakob Otto

**HAW Hamburg** 

23. April 2019



## Ablauf

#### Praktikum

- ▶ Was ist zu tun?
- ► Was braucht ihr?
- Beispielcode
- ▶ Tipps







# Aufgabenzettel





#### Was ist zu tun?

- DAC verstehen!
- Sinus/Sägezahnsignale ausgeben
- verschiedene Frequenzen darstellen
- verschiedene Amplituden darstellen





# Beispielcode





# Wie kommt ihr an Samples?!

Für die Lookup-tables braucht ihr Samples.

- volle Periode des Signals berechnen
- Samples in einem Array hard-coden
- $\bullet$  Am besten ohne Offset speichern  $\to$  Signal sollte um 0-pkt laufen.
- erst beim nutzen geeignet umformen.



## Wie Umformen?

- Samples z.B. in +1V/-1V Format speichern
- Beim nutzen dann teilen

```
// darstellung +1V/-1V
fifo[index] = samples[sampleIndex] + offset;
// Darstellung +0.5V/-0.5V
fifo[index] = (samples[sampleIndex] >> 1) + offset;
```

# Berechnungsbeispiel





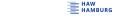
# Berechnung der Schrittweite

Zum Darstellen verschiedener Frequenzen benötigt ihr verschiedene Schrittweiten.

- ullet kleine Schrittweite o kleine Frequenz
- ullet große Schrittweite o große Frequenz

### Berechnung:

```
delta_{freq} = ((((ANZ\_SAMPLES) * FREQ) << frac)/TIMER\_FREQ)
frac = fractional Anteil des Q-Formats
```



Schrittweite  $\rightarrow$  Q-Format?!

Für höchste genauigkeit Q-Format nutzen!





## Schrittweite $\rightarrow$ Q-Format?!

Für höchste genauigkeit Q-Format nutzen!

Bei 360 samples brauchen wir 9 Integer-bits  $\rightarrow$  29 = 512

Qu9.23 ist also sinnvolles Format

