

# **Klärende Antworten und Fragen zu Aufgabe 2**

**Prof. Dr. Martin Becke**

Martin.Becke@haw-hamburg.de

## Bemerkungen zur Aufgabe 2

- Silke Behn zeigt wie unser Oszilloskop funktioniert!
- Wenn der Timer gestartet wird, und der PSC auf 0 steht (ist Standard), dann kann mit TIM2->CNT der Timer abgefragt werden.
  - Er läuft dann mit einer Geschwindigkeit von 90 MHz,
  - Erhöhung von TIM2-CNT alle 1/900000000s.
  - Timer vorher einschalten mit initTimer().
  - Overflow nach 0xFFFFFFFF (muss behandelt werden)
  - Eine 360° Drehung bedeutet 1200 Ticks, 1200 mal drehen.
  - Grad berechnen :  $360 * \text{meineTicks} / 1200 = 3/10 * \text{meineTicks}$

## LCD

- Ausgabe mit Integer-Arithmetik (Auf dem LCD Display):
  - `(int) (meineTicks*3/10)` .
  - `(int) (meineTicks*3%10)`
  - LCD sollte nicht flackern, kann man erreichen wenn nicht in jedem Schleifendurchlauf ausgegeben wird.
  - Beim Kompilieren keine „Warnings“
  - DDC sollte eingehalten werden

## Zustandsübergänge des Drehgebers

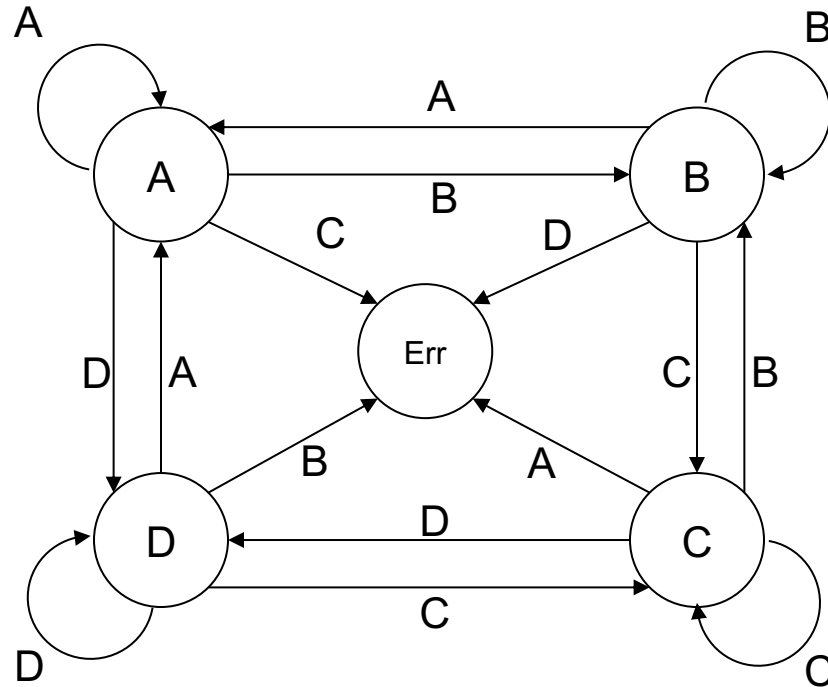
Zuordnung Signale - Phase		
Kanal A	Kanal B	Phase
0	0	a
1	0	b
1	1	c
0	1	d

Zuordnung Phasenwechsel - Ereignis	
Phasenwechsel	Ergebnis
Phase b → Phase a	Rückwärtslauf
Phase b → Phase b	Keine Änderung
Phase b → Phase c	Vorwärtslauf
Phase b → Phase d	Fehler

Sind alle Übergänge möglich?

Wie kann man einen Automaten / eine FSM einsetzen?

## Kodierung in einen Automaten



Bei den Übergängen muss beachtet werden, ob der Drehgeber links oder rechts herum dreht.

Startzustand?

## Kodierung eines Automaten über Switch Statement

- Switch Statement über die Zustände
- Pro Zustand Switch Statement über die Eingaben

# Kodierung eines Automaten über eine Tabelle

## ➤ Zweidimensionales Feld

```
#define PHASE_A      0x00    // A = 0 B = 0
#define PHASE_B      0x02    // A = 1 B = 0
#define PHASE_C      0x03    // A = 1 B = 1
#define PHASE_D      0x01    // A = 0 B = 1

// Automaton verwaltet die Phaseuebergänge
// Kodierung der Zustände : <letzte Phase><Drehrichtung>
typedef enum {Start=0, Err=1, AForRot=2, ABackRot=3, ANoRot=4, BForRot=5,
             BBackRot=6, BNoRot=7, CForRot=8, CBackRot=9, CNoRot=10,
             DForRot=11, DBackRot=12, DNoRot=13} StateType;
```

## Kodierung eines Automaten über eine Tabelle

```
static StateType delta[14][4] = {
    // PHASE_A    PHASE_D    PHASE_B    PHASE_C
    /* Start      */ {ANoRot,    DNoRot,    BNoRot,    CNoRot} ,
    /* Err        */ {Err,       Err,       Err,     Err } ,

    /* AForRot    */ {AForRot,    DBackRot,    BForRot,    Err } ,
    /* ABackRot   */ {ABackRot,    DBackRot,    BForRot,    Err} ,
    /* ANoRot     */ {ANoRot,     DBackRot,    BForRot,    Err} ,

    /* BForRot    */ {ABackRot,    Err,        BForRot,    CForRot } ,
    /* BBackRot   */ {ABackRot,    Err,        BBackRot,    CForRot} ,
    /* BNoRot     */ {ABackRot,    Err,        BNoRot,    CForRot} ,

    /* CForRot    */ {Err,        DForRot,    BBackRot,    CForRot } ,
    /* CBackRot   */ {Err,        DForRot,    BBackRot,    CBackRot} ,
    /* CNoRot     */ {Err,        DForRot,    BBackRot,    CNoRot} ,

    /* DForRot    */ {AForRot,    DForRot,    Err,        CBackRot } ,
    /* DBackRot   */ {AForRot,    DBackRot,    Err,        CBackRot} ,
    /* DNoRot     */ {AForRot,    DNoRot,     Err,        CBackRot} };

state = delta[state][input]
```



## Diskussion der Zeit

- Was bedeutet 100 to 360 cycles per revolution (CPR)
- Was bedeutet 400 to 1440 pulses per revolution (PPR)
- Kann man zu schnell am Drehgeber drehen?
- Woran erkennt man das?
- Kann man jeden Fehler erkennen?
- Gibt es zeitliche Schranke?