

Introduktion til State Machines

Automationsteknologuddannelsen

28. januar 2025

Hvad er en State Machine?

Definition:

- ▶ En state machine (tilstandsmaskine) er en model, der beskriver systemets tilstande og overgange mellem dem.
- ▶ Bruges til at definere, hvordan et system opfører sig baseret på input.

Anvendelse:

- ▶ Kontrolsystemer.
- ▶ Kommunikation mellem komponenter.
- ▶ Visualisering af systemets adfærd.

Typen af State Machines

To hovedtyper:

- ▶ **Moore-maskiner:** Output afhænger kun af den aktuelle tilstand.
- ▶ **Mealy-maskiner:** Output afhænger både af den aktuelle tilstand og input.

Eksempler:

- ▶ **Moore-maskine:** Automatisk lys, hvor tilstanden "Tænd" kun afhænger af, om det er mørkt.
- ▶ **Mealy-maskine:** Trafiklys, hvor lyset skifter afhængigt af timer (tilstand) og knaptryk (input).

Standarder for State Machines

Relevante standarder:

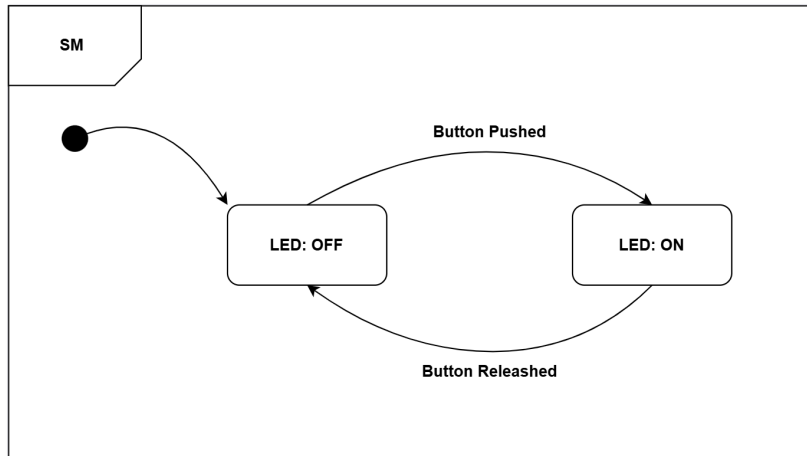
- ▶ **UML (Unified Modeling Language):** Standardiseret til at beskrive tilstande og overgange.
- ▶ **SysML (Systems Modeling Language):** Bruges til tværfaglige systemer og understøtter state machines.

UML State Machine Diagrammer:

- ▶ Repræsenterer tilstande, overgange og handlinger i komplekse systemer.
- ▶ Anvendes ofte i software- og systemdesign.

Eksempel #1 på en State Machine

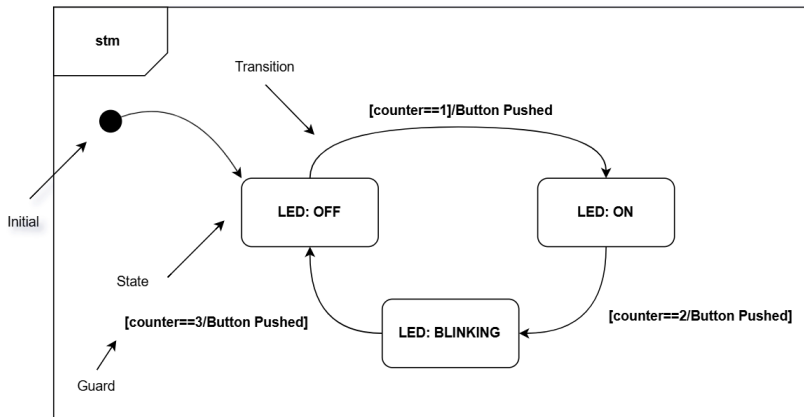
Eksempel: LED-styring



Figur: Tilstandsmaskine for LED-styring.

Eksempel #2 på en State Machine

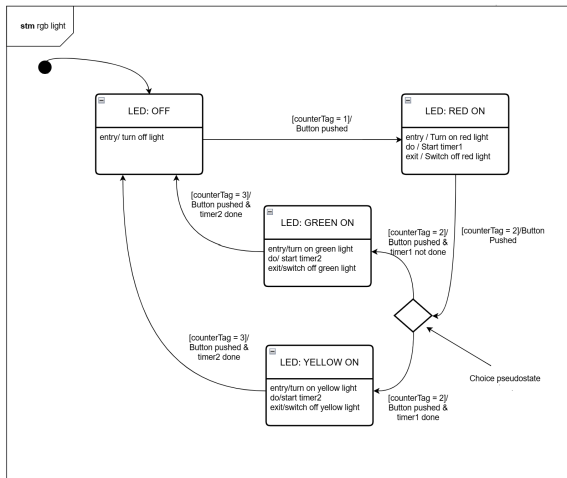
Eksempel: LED-styring



Figur: Tilstandsmaskine for Blinking LED-styring.

Eksempel #3 på en State Machine

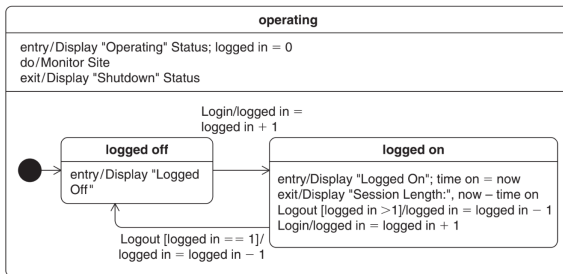
Eksempel: RGB-styring



Figur: Tilstandsmaskine for RGB-styring.

Eksempel #4 på en State Machine

Eksempel: RGB-styring



Figur: Tilstandsmaskine for nested system.

Sådan opbygger du en State Machine

Trin-for-trin vejledning:

1. Definér systemets mulige tilstande (fx "Tændt", "Slukket").
2. Identificér overgange mellem tilstande og deres betingelser.
3. Tegn diagrammet med tilstande som cirkler og overgange som pile.
4. Beskriv handlinger, der udføres i hver tilstand eller overgang.

Tips:

- ▶ Hold diagrammet overskueligt.
- ▶ Brug klare labels til overgange og tilstande.

Typiske fejl og faldgruber

Fejl:

- ▶ Manglende beskrivelse af betingelser for overgange.
- ▶ For mange tilstande, der gør diagrammet komplekst.
- ▶ Uklar eller manglende labeling.

Sådan undgår du fejl:

- ▶ Begræns antallet af tilstande til det nødvendige.
- ▶ Brug ensartede labels og navngivning.
- ▶ Test, at alle mulige scenarier er dækket.

Opgave:

- ▶ Lav en state machine for en trafiklys-kontrol.
- ▶ Tilstande:
 - ▶ Rødt lys.
 - ▶ Gult lys.
 - ▶ Grønt lys.
- ▶ Overgange: Definér tid eller knaptryk som betingelser.

Tips:

- ▶ Start med få tilstande og tilføj gradvist kompleksitet.
- ▶ Brug UML eller SysML-standarder for at sikre klarhed.

Husk:

- ▶ En god state machine er simpel, logisk og dækker alle mulige tilstande.
- ▶ Brug state machines til at forstå og dokumentere komplekse systemer.

Videre arbejde:

- ▶ Forbind state machines med flowcharts og blokdiagrammer for en helhedsforståelse.
- ▶ Brug state machines aktivt i projektstyring og fejlfinding.