

# Overkørsel st. enkeltsporet strækning

---

Type	Specifikation	Oprettet	07-04-2021
Forfatter	Jan Birch	Rettet	29-04-2021
Noter:			

## Indledning

En model for en overkørsel bliver bygget med Arduino på et breadboard.

Overkørslen bliver konfigureret i `setup()`. Hovedprogrammet `loop()` indeholder en simpel algoritme for overkørsel.

## Arduino tilpasning

Arduino har ikke et operativsystem. Program skal levere al maskinkode. Der er begrænset memory til data 2kbyte og 32kb til program.

Memory allokering og deallokering kan udvide brug af memory, men indfører risiko for at fylde memory op med døde data. I dette projekt bliver al brug af memory lagt fast på compile tidspunktet. Når der kun afsættes statisk memory får det som konsekvens:

- Lister med variabelt antal elementer, må lægges i et array variabel med fast og maksimal længde.  
Funktionen `sizeof(array[0])` returnerer antal elementer i et array med en given datatype.
- Operatorerne `new` og `delete` kan ikke bruges.

Normalt håndteres tryk på en knap og udløb af en timer med interrupt. Det er en kompleks mekanisme, der skal programmeres og testes omhyggeligt, for at forebygge runtime fejl. I dette projekt bliver polling brugt. Tidsinterval for polling bliver så kort, at f.eks. et knaptryk bliver besvaret hurtigt nok.

Fejlhåndtering med exceptions og `throw` er også en kompleks mekanisme, der så vidt vides ikke understøttes af arduino. Softwarefejl må forebygges med grundig testning.

## Hardware

Hardware består af Arduino, eksterne enheder der giver input og eksterne enheder der modtager output.

Input og outputenheder bliver koblet til en digital port med parallel kommunikation.

## Sensorer og knapper

Der er en række enheder der kan levere input, hver med deres egenskaber.

- Trykknop. En trykknop leverer en impuls.

## Signaler, klokker og bomme

Der er en række enheder der simulerer output.

- LED som kan være: Slukket, tændt med fast lys eller blinke.
- Buzzer som kan være: Slukket, tændt med vedvarende lyd eller give pulserende lyd.

# Overkørsel st. enkeltsporet strækning

---

## Tilslutning pin

Arduino får følgende ind- og udgange:

Funktion	Pin
Reserveret seriel kommunikation	0
Reserveret seriel kommunikation	1
Indgang: Knap for manuel tænd og sluk	2
N.A.	3
N.A.	4
N.A.	5
N.A.	6
Udgang køreretning AB: Uordenssignal gul LED	7
Udgang køreretning BA: Uordenssignal gul LED	8
N.A.	9
Udgang vejklodder: Aktiv buzzer	10
Udgang vejlys: Rød LED	11
N.A.	12
N.A.	13

## Software

### Klasser

En klasse skal ifølge god praksis for objektorienteret programmering, have ansvar for og håndtere 1 bestemt opgave.

Software bliver bygget med komponenter. På den måde kan der hen ad vejen programmeres nye komponenter, som helt enkelt kan indgå i en overkørsel. Ved programmeringen kan komponenter testes i et simpelt testprogram, inden komponenten sættes ind i programmet til overkørsel.

Opbygning af en konkret overkørsel skal være let at forstå. Det bliver den ved at.

1. De komponenter overkørslen skal bestå af bliver udvalgt.
2. Komponenter bliver koblet sammen.
3. Softwaren udfører den interne proces.

Generelt når der er behov for at konfigurere et objekts medlem får metoden navnet setXyz. Tilsvarende ved aflæsning af et medlem getXyz. Xyz er medlemmets navn.

### Drivere

Input og output kan udveksles med hardware via parallel driver.

En driver bliver en eksakt model af hardware. F.eks. for en knap, så melder driver tændt, når der trykkes på knappen og meldt slukket, når knappen er sluppet. En driver for en knap skal have indbygget en venteperiode for kontaktprel. Så tilstand først meldes når kontakten er stabil.

I software bliver der brugt en navngivning for driverens tilstand: High og Low.

I software bliver metoder navngivet: Read og write.

Der bliver behov for følgende drivere:

- Trykknop. Den giver en impuls. En knap er enten normally high eller normally low.
- LED eller buzzer. Simpelt tænd eller sluk.

# Overkørsel st. enkeltsporet strækning

---

Der er en detalje i styringen. Hvis en Arduino port bliver sat ved hver polling, så kan den ikke følge med. Porten må kun blive sat ved en aktuel ændring. Det skal driver sørge for.

En parallel driver bliver knyttet til en Arduino port.

Se i øvrigt softwarepattern facade.

## Tidsstyring

Arduino har en klokfrekvens på 16MHz og 32kb programmemory. Ved 32kb programmemory får hver adresse 16bit eller 2byte. Det vil sige med 16MHz kan arduino gennemløbe programmemory på 1msek. Dertil kommer adressering af datamemory. Den mindste periodetid er ca. 2msek. Hele softwaremaskinen styres af polling med en periodetid på 5msek. Det er hurtigt nok til findeling af timeres periode. Alle timere kan blive simple tællere.

## Timer

Mange komponenter får brug for en timer.

- Ved oprettelse kan den få en varighed.
- Undervejs kan konfigureres en varighed.
- Når tid er udløbet, returneres en "triggered".

## Ur

Uret varetager klokkecyklus.

## Blinker

Blinker kører konstant med fast periodetid.

Blinker skabes globalt, da der kun skal være 1.

Der skal være funktioner:

- Bool blinkerSubscriber. Funktionen returnerer impuls fra blinker. Alle brugere får en pointer til funktion. Den indkapsler funktionen. p\_ blinkerSubscriber. Funktionspointer til blinkerSubscriber.

Se i øvrigt softwarepattern observer.

## Tilstandsmaskine

Tilstandsmaskine bygges simpelt med lister:

- Over mulige tilstande.
- Tider for opløsning af sikret, vejsignalering og billisttid.

Tilstandsmaskine får sin egen timer.

## Konfiguration og opstart

Alle nødvendige konstanter og enum specificeret globalt. Objekter bliver instantieret globalt.

Drivere bliver konfigureret ved instantiering.

I setup() bliver start tilstand specificeret.

Hovedprogrammet loop() udfører tilstandsmaskinen.

En tilstand sørger for blink i vejlys og -klokker (buzzer simulerer lyden af tændt og slukket).