Overkørsel st. enkeltsporet strækning

Type	Specifikation	Oprettet	07-04-2021
Forfatter	Jan Birch	Rettet	09-05-2021
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.

- Trykknap, der leverer en impuls.
- Omskifter, der står i en fast stilling.
- Seriel kommunikation. Kommunikationen leverer en adresse og en impuls.

Signaler og klokker
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.

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 vejklokker: Aktiv buzzer	10
Udgang vejlys: Rød LED	11
N.A.	12
N.A.	13

Software

Klasser

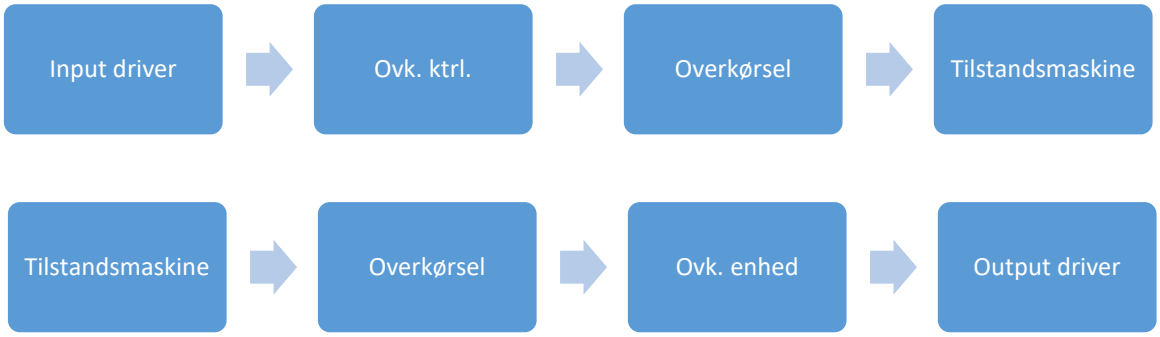
En klasse skal ifølge god praksis for objektorienteret programmering, have ansvar for og håndtere 1 bestemt opgave.

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.

Proces for en overkørsel bliver:



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.

Overkørsel st. enkeltsporet strækning

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:

- Trykknap. Den giver en impuls. En knap er enten normally high eller normally low.
- Omskifter. Den giver konstant input og er binær. En omskifter er normally high eller normally low. Dette er en fremtidig mulighed, endnu ikke indbygget.
- LED eller buzzer. Simpelt tænd eller sluk.

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.

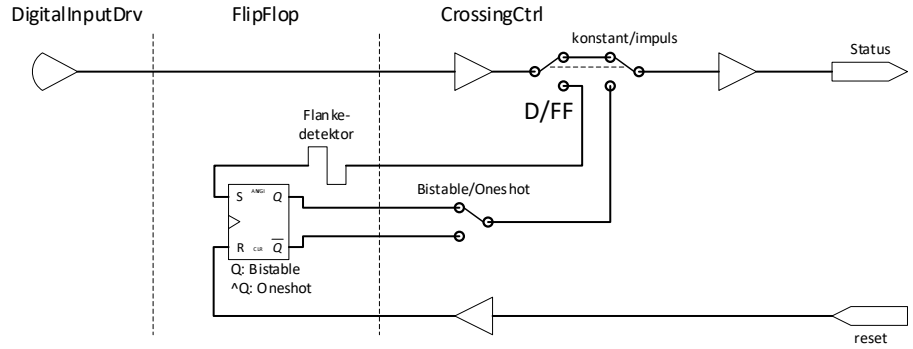
Se i øvrigt softwarepattern facade.

Overkørsel betjening og sensor (kontrol)

Overkørsel kontrol skal kunne konfigureres til den aktuelle anvendelse: Manuel tænd og sluk, togvej eller tændsted.

- Fra en omskifter skal input fra driver leveres videre uændret til kontrollens udgang.
- Fra en driver der giver en impuls skal kontrollens udgang holdes fast indtil ny impuls kommer eller indtil reset.

Tegningen herunder viser i store træk funktionen.



En kontrol kobler driver og evt. hukommelse til overkørslens tilstandsmaskine.
Et magasin bruger en flipflop som hukommelse.

En kontrol får som funktion.

- Kontrol får tilknyttet en driver.
- En driver leverer enten høj eller lav, når den er aktiv.
- Kontrol kan kobles til en flipflop og konfigueres til enten bistabil eller oneshot.
  Oneshot bruges af tændsted.
- Når der kobles en flipflop på, så skal den interne logik automatisk stille om fra konstant til impuls styret.

- Kontrol modtager polling. I et gennemløb handles der på driverens tilstand.
- Kontrol skal sende reset videre til flipflop.
- Kontrol skal kunne levere sin værdi.

I software bliver der brugt en navngivning:
- Udgangens tilstand: On eller Off.
- Valg af flipfloptype: Bistable og Oneshot.

I software bliver metoder navngivet: status, reset.

Flipflop
En driver kan gå enten høj eller gå lav, når den bliver aktiv. FlipFlop skal kunne konfigureres til den aktuelle type driver der er sat op til normally open eller normally closed. Konfigurationen sørger for at flipflop står i korrekt start tilstand.
Når flipflop er aktiv er udgangen on. Flipflop skal kunne modtage reset og gå off.

I software bliver der brugt en navngivning:
- Flipflop indstilles til normally high eller normally low.
- Udgangens tilstand: On eller Off.

Overkørsel ydre enheder
Overkørsel bliver simuleret med følgende enheder.
- Uordenssignal. Signalet er enten tændt eller slukket.
- Overkørselssignal. Beskrivelse venter til senere.
- Vejlys. Signalet er enten tændt med blinkende lys eller slukket.
- Vejklokker. Klokken er enten slukket eller tændt med pulserende lyd.
- Vejbomme med servomotor. Bomme oppe vist med oppe stilling. Bomme nede bliver vist med nede stilling. Bomme ned bliver vist med bevægelse fra op til ned. Bomme op bliver vist med bevægelse fra ned til op.

En overkørsel ydre enhed bliver sat i tilstand: Block eller Pass.

Internt i en overkørsels ydre enhed bliver der brugt en navngivning for sekvens:
On, Off, Blink, BarrierUp, BarrierGoDown, BarrierDown, BarrierGoUp.

En overkørsel ydre enhed får som funktion.
- Enheden modtager besked om ny tilstand.
- Enheder der skal kunne vise blink eller pulserende lyd, skal modtage polling.
- Enheden til vejbom, må omregne tid til grader bombevægelse. Bommen skal bevæges så glidende som muligt og indenfor den specificerede tidsperiode. Enheden skal modtage polling.

I software bliver metoder navngivet: to.

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).