

Overkørsel st. enkeltsporet strækning

Type	Design	Oprettet	02-04-2021
Forfatter	Jan Birch	Rettet	21-05-2021
Noter:			

Indledning

En model til styring og overvågning af en overkørsel er bygget til Arduino og tilsluttet med elektronik.

Modellen indeholder en samling af komponenter og mekanismer. De kommunikerer udelukkende med meddelelser. Hovedprogrammet `setup()` kobler alle komponenter og mekanismer sammen. Det vil sige en overkørsel er konfigureret i `setup()`. Hovedprogrammet `loop()` og alle klasser er færdigbygget og uden behov for modifikation. Nye komponenter kan tilføjes. Det kan f.eks. være et tændsted, seriel I/O med mere.

Arduino tilpasning

Arduino Uno 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 denne model er alt memory lagt fast på compile tidspunktet. Når der kun afsættes statisk memory får det som konsekvens:

- Alle objekter bliver instantieret.
- Antal elementer i et array bliver lagt fast.

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. Denne model bruger polling. 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, betjeningsenheder og sensorer der giver input og ydre enheder der modtager output.

Sensorer og knapper

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

- Trykknop, der leverer en impuls.
- Omskifter, der står i en fast stilling.
- Kommunikation med sikringsanlæg og andre overkørsler. Det er ikke indbygget i model.

Signaler, klokker og bomme

Der er en række enheder der simulerer overkørselens ydre enheder.

- LED som kan være: Slukket, tændt med fast lys eller blinke.
- Buzzer som kan være: Slukket, tændt med vedvarende lyd eller give pulserende lyd.
- Servomotor der kan være: Stå i en fast stilling, bevæge sig med stigende vinkel eller med faldende vinkel.

Overkørsel st. enkeltsporet strækning

Tilslutning pin

For en Arduino UNO kan følgende ind- og udgange bruges:

Funktion	Reserveret	Pin
Reserveret seriel kommunikation	X	0
Reserveret seriel kommunikation	X	1
Indgang: Knap for manuel tænd og sluk	Valgfri	2
Indgang: Knap for simulering af tændsted køreretning AB	Valgfri	3
Indgang: Knap for simulering af tændsted køreretning BA	Valgfri	4
Indgang: Knap for simulering af togvej sat	Valgfri	5
Udgang vejbmme: Reserve	X	6
Udgang køreretning AB: Uordenssignal gul LED	Valgfri	7
Udgang køreretning BA: Uordenssignal gul LED	Valgfri	8
Udgang vejbmme: Servomotor	X	9
Udgang vejkløkker: Aktiv buzzer	Valgfri	10
Udgang vejlys: Rød LED	Valgfri	11
Udgang: Grøn LED for togvej	Valgfri	12
		13

Pin mærket med X kan bruges til andet formål. Risiko er at det spærrer for fremtidig nyudvikling.

Når der bygges en overkørsel, kan pin vælges. Slå op i Arduino dokumentation og find de porte der kan bruges.

Kommunikation

Model er bygget til parallel kommunikation.

Seriel kommunikation foregår typisk ved at en enhed får en adresse, som klistres på en kommunikationspakke. Seriel kommunikation er ikke undersøgt i detaljer.

Model er forberedt til seriel kommunikation. For at opnå seriel kommunikation, skal der tilføjes komponenter til model, som kobles på med pointere og håndteres med polymorfisme.

Software

Klasser

En klasse følger god praksis for objektorienteret programmering, den har 1 ansvar og håndterer 1 bestemt opgave.

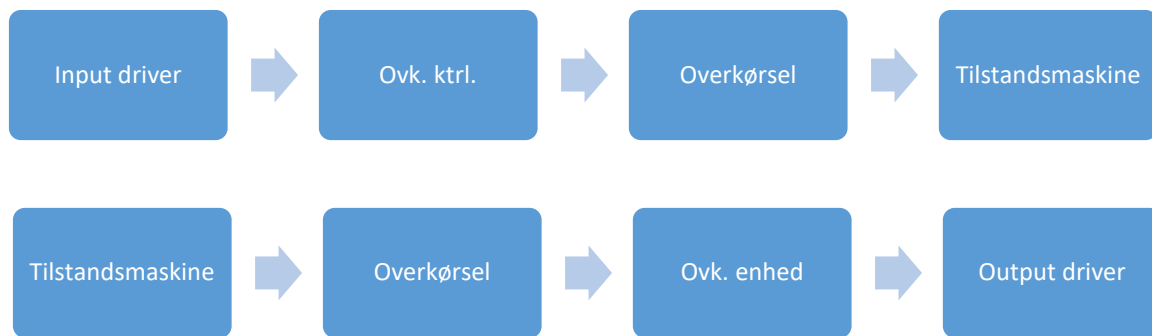
Software er bygget med komponenter. På den måde kan der hen ad vejen programmeres nye komponenter, som helt enkelt kan indgå i en overkørsel. Ved programmeringen kan komponenter testes i et simpelt testprogram, inden komponenten sættes ind i modellen til overkørsel.

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:

Overkørsel st. enkeltsporet strækning



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

Drivere

Input og output kan udveksles med hardware enten via parallel eller seriel kommunikation. Den konkrete overkørsel må være fleksibel, så enten en parallel eller en seriel driver kan kobles på.

En driver er 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 har indbygget en venteperiode for kontaktprel. Så tilstand først meldes når kontakten er stabil.

Driverens tilstand meldes med: High og Low.
I software er metoder navngivet: Read og write.

Der er indbygget følgende drivere:

- Trykknop. 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. Den bruger samme klasse som trykknop.
- LED eller buzzer. Simpelt tænd eller sluk.
- Vejbom. Driver sørger for en glidende bevægelse af bom. Den skal konfigureres med antal grader / sekund, som tilpasses så servomotor ikke trækker for stor strøm.

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 sørger driver for.

En parallel driver bliver knyttet til en Arduino port ved opsætning.
En seriel driver bliver knyttet til flere Arduino porte ved opsætning.

Se i øvrigt softwarepattern facade.

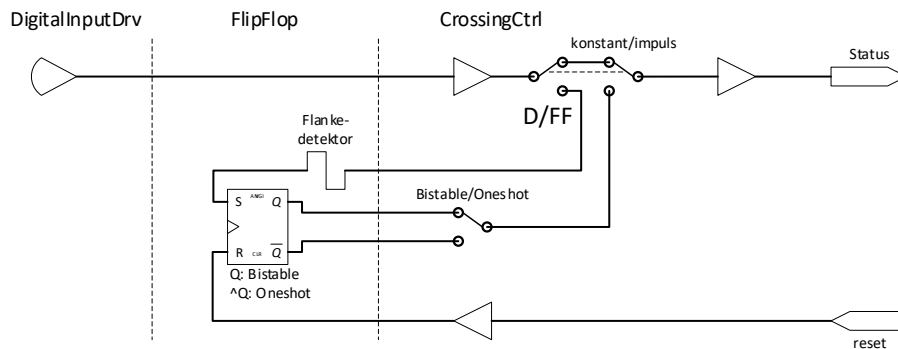
Overkørsel betjening og sensor (kontrol)

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

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

Tegningen herunder viser i store træk funktionen.

Overkørsel st. enkeltsporet strækning



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

En kontrol har som funktion.

- Kontrol får tilknyttet en driver ved opsætning.
- En driver leverer enten høj eller lav, når den er aktiv.
- Kontrol kan kobles til en flipflop og konfigureres til enten bistabil eller oneshot. Oneshot bruges af tændsted.
- Når der kobles en flipflop på, så omstiller den interne logik automatisk om fra konstant til impuls styret.
- Kontrol modtager polling. I et gennemløb handles der på driverens tilstand.
- Kontrol sender reset videre til flipflop.
- Kontrol leverer sin status.

I software er der brugt navngivning:

- Udgangens tilstand: On eller Off.
- Valg af flipfloptype: Bistable og Oneshot.

I software er metoder navngivet: status, reset.

Flipflop

En driver kan gå enten høj eller gå lav, når den bliver aktiv. FlipFlop kan 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 kan modtage reset og gå off.

En flipflop starter med off.

I software er der brugt navngivningen:

- Flipflop indstilles til normally high eller normally low.
- Udgangens tilstand: On eller Off.

Overkørsel st. enkeltsporet strækning

Overkørsel ydre enheder

Overkørsel bliver simuleret med følgende enheder.

- Uordenssignal. Signalet er enten tændt eller slukket.
- Overkørselssignal. Signalet er enten tændt eller slukket.
- Vejlys. Signalet er enten tændt med blinkende lys eller slukket.
- Vejklokke. Klokken er enten slukket eller tændt med pulserende lyd.
- Vejbom med servomotor. Som enten skal være oppe eller nede.

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

En overkørsel ydre enhed har som funktion.

- Enheden modtager besked om ny tilstand.
- Enheder der skal kunne vise blink eller pulserende lyd, modtager polling.
- Enheden til vejbom, styrer driver til vejbom, som får besked om bom op eller bom ned. Enheden modtager polling.

I software er metoder navngivet: to.

Tilstandsmaskine: stateMachine

Den grundlæggende logik er en state machine. Se state softwarepattern.

Tilstandsmaskinen kommunikerer via overkørsels objektet med kontroller og ydre enheder. Det giver en fleksibel mulighed for, at vælge hvilke komponenter, der skal kobles på den aktuelle overkørsel.

Til en konkret overkørsel skal der programmeres en subklasse per state som.

- Henter en meddelelse fra overkørsel kontrol:
 - Tjekker om betingelser for transition er opfyldt.
 - Udløser transition til næste tilstand.
- Sender meddelelse til overkørsel ydre enhed om tilstand (block eller pass).
- Skal kunne resette overkørsel kontrol.
- Kan slutte sig til en timer, som returnerer udløb af tid, der udløser en transition. Tidsforsinkelse skal konfigureres.

Overkørsel

Der er mange objekter der skal have forbindelse til hinanden. Der er et problem med objektorienteret kode. Objekt A skal referere objekt B. Objekt B skal referere objekt A. Det giver vanskeligheder fordi compiler ikke kender klassen B, når specifikation for A bliver læst. Rækkefølgen af specifikationer i koden bliver kritisk. Til denne software bruges et design med referencer.

- Overkørsel kontrol og ydre enhed referer til input- og outputdriver.
- Overkørsel ydre enhed refererer til blinker.
- Overkørsels objekt referer kun til tilstandsmaskinens grænseflade.
- Det er afledte tilstande, der refererer til overkørsels objekt.
- Overkørselsobjekt refererer til en samling med pointere til kontroller, ydre enheder og afledte tilstande.
- Polling refererer til overkørsel og blinker.

Så går referencer kun 1 vej.

Alle overkørsels kontroller er samlet i et array[], som får tildelt memory på compile time.

Indeks i array er en simpel enum for eksempel: TaendSluk, Togvej, Taendsted.

Overkørsel st. enkeltsporet strækning

Hver overkørsel kontrol har en fast plads, som gør opslag i samlingen enkel.

Alle overkørsels overkørsel ydre enheder er samlet i et array[], som får tildelt memory på compile time. Indeks i array er en simpel enum for eksempel: OUSignalA, OUSignalB, VejSignal, Vejklokke, Vejbom

Hver overkørsel ydre enhed har en fast plads, som gør opslag i samlingen enkel.

Når der bruges et array til kontroller og ydre enheder bliver de uafhængige af tilstandsmaskine. Løsningen giver mulighed for i fremtiden at koble nye kontroller og ydre enheder på.

Alle overkørsels afledte tilstande er samlet i et array[], som får tildelt memory på compile time. Indeks i array er en simpel enum for eksempel: Ikke sikret, forringning, Sikret, Opløsning, Billisttid. Hver tilstand har en fast plads, som gør opslag i samlingen enkel.

Se i øvrigt softwarepattern mediator.

Tidsstyring

Arduino har en klokkefrekvens 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 er 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 ligger globalt, da der kun skal være 1.

Der er følgende funktioner:

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

Et overkørsel ydre enheder som bruger blinker har indbygget konfiguration af sin egen pointer til blinkerSubscriber. Indbygning er i constructor, så referencen bliver synlig.

Se i øvrigt softwarepattern observer.

Polling

Arduino loop() udfører polling.

1. Kalder pendulum.

Overkørsel st. enkeltsporet strækning

2. Poller blinker.
3. Poller overkørsel, som:
 - a) Poller alle overkørsel kontroller.
 - b) Poller alle overkørsel ydre enheder.
 - c) Kører tilstandsmaskine.

Konfiguration og opstart

Alle objekter bliver instantieret i global scope, så Arduino får fastlagt memory i compile time.

En række software elementer bruges internt og får tildelt variabelnavn som vist i tabellen herunder.

Klasse	Objektets variabelnavn
DigitalInDrv	Ingen kun abstrakt klasse
DigitalOutDrv	Ingen kun abstrakt klasse
CrossingCtrl	Specifikt ved indbygning
CrossingDevice	Ingen kun abstrakt klasse
StateMachine	stateMachine
Clock	Specifikt ved indbygning

Alle nødvendige konstanter og enum er specificeret globalt.

Kobling til hardware og initialisering af objekter sker i konstruktor.

I setup() bliver overkørsel konfigureret.

1. Hver driver bliver knyttet til sin overkørsel kontrol eller ydre enhed.
2. Hver overkørsel kontrol bliver knyttet til overkørsel.
3. Eventuel flipflop bliver koblet på overkørsel kontrol.
4. Hver overkørsel ydre enhed bliver knyttet til overkørsel.
5. Hvert tilstandsobjekt bliver meldt ind i tilstandsmaskinen.
6. Overkørsel bliver sat i start tilstand.