

Overkørsel st. enkeltsporet strækning

Type	Specifikation	Oprettet	07-04-2021
Forfatter	Jan Birch	Rettet	17-04-2021
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. En Arduino port konfigureres til enten at være en indgang eller en udgang. En indgang konfigureres til enten at være høj eller lav, når knap er passiv.

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.

Tilslutning pin

Arduino får følgende ind- og udgange:

Overkørsel st. enkeltsporet strækning

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

Polling

Arduino loop() sætter en ny cyklus i gang ved at udføre polling.

Trykknop

Kobling til hardware sker via en driver: PushButton

Driver er et eksakt spejl på hardware indgangens tilstand.

Trykknop modtager en betjening. Når knappen bliver sluppet, går enhed tilbage til hviletilstand.

Hardware tilslutning "normally open eller normally closed" bliver konfigureret.

En digital port der trukket høj internt af Arduino, er "normally closed".

Signaler og vejklodder

Kobling til hardware sker via en driver: SimpleOnOff

Driver er et eksakt spejl på hardware udgangens tilstand.

LED eller buzzer kan blive tændt eller slukket.

Blinker

Blinker er en funktion med static variable. Funktionen leverer tændt eller slukket ud fra en periodetid. Periodetid er for blink til vejlys, klodder med mere. Funktionen initialiseres i setup.

Timere

Timere bliver indbygget fra biblioteket BlockNot.

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.

Overkørsel st. enkeltsporet strækning

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).