

Betjener

31. januar 2023 15:17

Betjener simulerer elektronikkomponenter, der bruges manuelt: Knapper, omskiftere, potentimeter m.m.

Biblioteket har grænseflade til applikation og kun få generelt brugbare softwarekomponenter. Applikationer skal bruge biblioteket til grænseflade og koble egne specifikke softwarekomponenter på det.

Bibliotek hedder: JBManual.h.

Funktion

En betjener modtager input fra en port i en input-driver. Porten bliver overvåget for hver klokkecyklus.

En betjener konfigureres til den aktuelle anvendelse. For eksempel:

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

En betjener har som funktion.

- Betjener får tilknyttet en driver og en port ved opsætning.
- En port leverer enten høj eller lav, når den er aktiv.
- En betjener kan kobles til en eller flere digitale funktioner.
- En betjener modtager polling. I et gennemløb handles der på portens tilstand.
- En betjener sender værdi fra port videre til digitale funktioner.
- En betjener sender reset videre til digitale funktioner.
- En betjener leverer sin status.
- I software er der brugt navngivning: Udgangens tilstand: ON eller OFF.

Opsætning

I hovedprogram skal der erklæres globalt:

- MaxNoManuals: unsigned int const=nr: Antal betjener i applikationen. Navn på konstant er obligatorisk.
- Enum-liste med betjenerenes navne.

I hovedprogram setup() skal der opsættes:

- <navn på betjener>.begin(...)

Grænseflade

Grænsefladen giver en ensartet tilkobling af betjener.

For grænsefladens design se softwarepattern facade.

Klasse

Klassens navn: t_Manual

Medlemmer

Grænsefladens medlemmer.

Navn	Type	Egenskab	Beskrivelse
driver	*InputDriver		Pointer til input drivere
portNo	unsigned int		Input portens nummer
state	byte		Betjenerens tilstand.

Metoder

Grænsefladens metoder.

Navn	Argumentliste	Returnerer	Beskrivelse
Constructor			Sætter default driver til ikke initialiseret. Sætter state default=OFF
begin	(driver: *InputDriver, portNo: unsigned int)		Betjening initialiseres.
doClockCycle	()		Overvåger port, for at holde digitale funktioner opdateret.
status	()	byte	Udlæser resultat af betjening.
reset	()		Resetter digitale funktioner og genlæser input port.

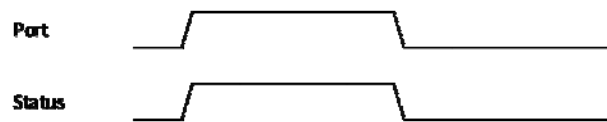
Simple knap

31. januar 2023 15:08

I en række applikationer er der behov for en knap, der blot følger om knappen er aktiv eller passiv. Brug af denne softwarekomponent gør det enkelt at tilslutte simple knapper.

Funktion

Knappen følger simpelt porten om den er HIGH eller LOW.



En simpel knap er kun forbundet med sin driver.



Opsætning og brug i komponent

En komponent, der skal bruge en betjening skal opsætte:

- Pointer til <navn på simpel knap>.

Klasse

Klassens navn: `t_SimpleButton`

Medlemmer

En simpel knap har ingen medlemmer.

Metoder

Den simple knaps metoder.

Navn	Argumentliste	Returnerer	Beskrivelse
<code>doClockCycle</code>	<code>()</code>		Input driver bliver aflæst i hver klokkescyklus.

Knap

1. februar 2023 13:03

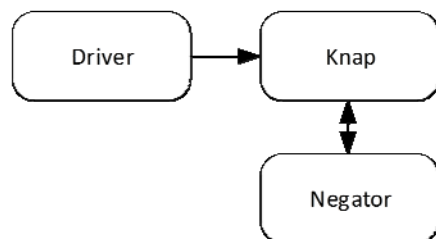
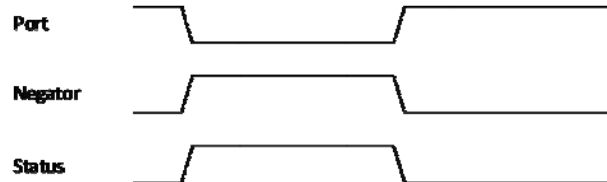
Knap simulerer de oftest brugte funktioner ved brug af trykknop og omskifter.

Funktion

En knap kan kobles til digitale funktioner, der giver fleksibilitet med at tilpasse funktionen af knappen.

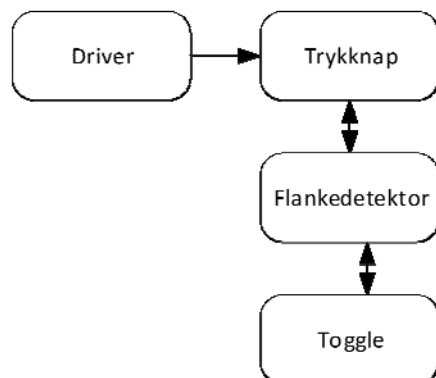
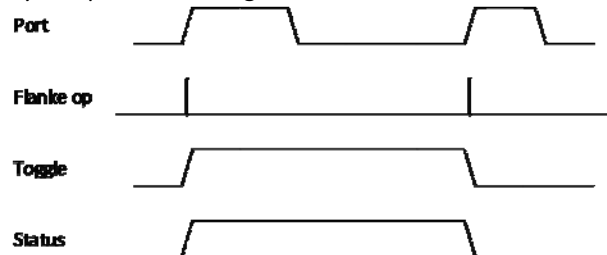
Eksempler:

Knap hvor input-port normalt er HIGH og hvor LOW er farligste stilling.



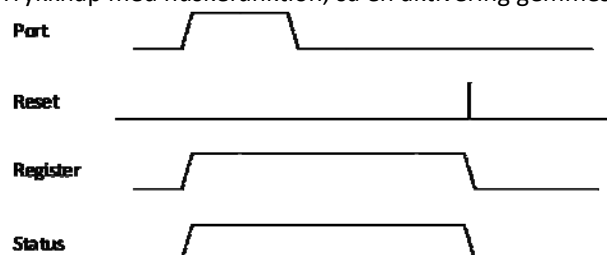
Negation konverter det binære signal til positiv logik.

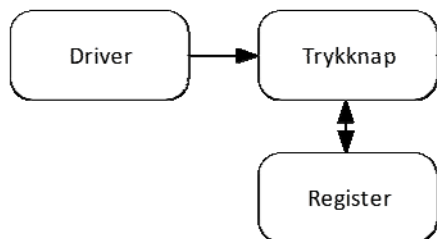
Trykknop med tænd og sluk funktion.



Flankedetektor sørger for at kun når input-port skifter til HIGH, kommer der signal ind. Toggle husker betjeningsens stilling indtil trykknappen igen bliver aktiveret.

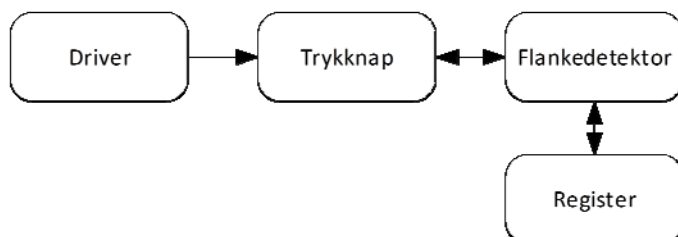
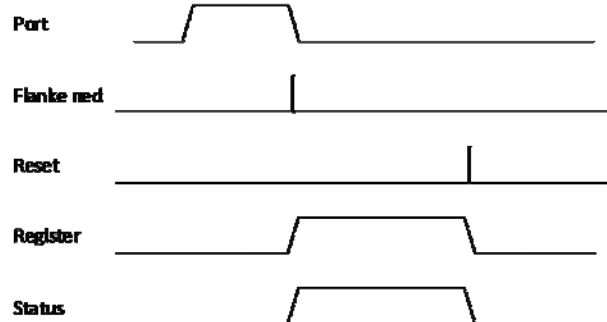
Trykknop med huskefunktion, så en aktivering gemmes til senere.





Register gemmer betjeningens stilling, indtil applikation sender et reset.

Trykknop med huskefunktion, som er koblet til en port, når den går LOW. Det vil sige når trykknop bliver sluppet.



Flankedetektor sørger for at kun når input-port skifter til LOW, kommer der signal ind. Betjeningen tjekker resultatet fra flankedetektor og når der er kommet HIGH ind, opdateres register. Register gemmer betjeningens stilling, indtil applikation sender et reset.

1 digital funktion kan kobles til en knap. Det er valgfrit om den bruges.

Knap understøtter ikke mere avancerede funktioner for eksempel langt tryk. Opstår den slags behov, skal der programmeres en ny softwarekomponent.

Opsætning og brug i komponent

En komponent, der skal bruge en betjening skal opsætte:

- Pointer til <navn på knap>.

Klasse

Klassens navn: t_Button

Medlemmer

Knappens medlemmer.

Navn	Type	Egenskab	Beskrivelse
digitalFunction	*DigitalFunction		Pointer til en digital funktion

Metoder

Knappens metoder.

Navn	Argumentliste	Returnerer	Beskrivelse
setDigitalFunction	(digitalFunction: *DigitalFunction)		Kobler digitalfunktion til betjening.
doClockCycle	()		Input driver bliver aflæst i hver klokkescyklus. Digitale funktioner bliver opdateret.
reset	()		Resetter digital funktion. Status bliver opdateret i næste klokkescyklus.