

# Digitale funktioner

7. januar 2023 21:32

Digitale funktioner er komponenter, der kan kobles på applikationens betjening, sensor simulatorer og styreenheder. Digitale funktioner behandler 1 bit.

Biblioteket hedder JBDigitalFunctions.h.

## Kobling af digitale funktioner

Et komponent i applikationen kobler sin digitale funktionskomponent ved hjælp af en pointer. En digital funktionskomponent har også en pointer, som kan bruges til at koble endnu en digital funktion på.

Det giver en række muligheder for at koble digitale funktioner på en applikationskomponent.

Applikationskomponent -> DigitalFct1 -> DigitalFct2

Eller

Applikationskomponent -> DigitalFct1  
-> DigitalFct2

Behandling af data bliver udført således. Vist ved en kædekoblet algoritme:

Trin	Applikation	DigitalFct1	DigitalFct2
1	VærdiA		
2	DigitalFct1->dataOut(VærdiA)	Værdi1 beregnes	
3		DigitalFct2->dataOut(Værdi1)	Værdi2 beregnes
4	VærdiA=Værdi2	<- Værdi2	<- Værdi2

Applikationskomponenten kan bruge kædekoblede komponenter, hvis slutresultatet alene skal bruges.

Har applikationskomponenten brug for mellemresultat fra DigitalFct1, skal de to digitale funktioner kaldes enkeltvis.

## Klasse

Klassens navn: t\_DigitalFunction

## Medlemmer

Grænsefladen har en række medlemmer fælles for alle digitale funktioner.

Navn	Type	Egenskab	Beskrivelse
value	bool		Værdien af den digital funktion
digitalFunction	*digitalFunction		Peger på næste digitale funktion

## Metoder

Standard metoder for en digital funktion.

Navn	Argumentliste	Returnerer	Beskrivelse
dataOut	(value: bool)	bool	Udlæser resultat af den digitale funktion
reset	()		Resetter den digitale funktion
setDigitalFunction	(nextDigitalFunction: *digitalFunction)		Sætter pointer til næste digitale funktion.

# Negator

22. januar 2023 22:01

Funktionens opgave er at levere 1bit negeret.

Negator kan for eksempel bruges efter et input, som er aktiv LOW og hvor den efterfølgende logik skal bruge ON=1.

## Funktion

Input	dataOut
0	1
1	0

Input og output bliver præsenteret af OFF og ON. Hvor ON = 1.

## Klasse

Klassens navn: t\_Negator

## Medlemmer

Negator arver sine medlemmer.

## Metoder

Negators metoder.

Navn	Argumentliste	Returnerer	Beskrivelse
dataOut	(value: bool)	bool	Udlæser resultat af input

# Flankedetektor

22. januar 2023 22:02

En flankedetektors opgave er at levere ON, når betingelser for et signals flanke er opfyldt.

## Funktion

Hver gang flankedetektor får nyt input, skal den genberegne output.  
Internt gemmer flankedetektor sin værdi fra forrige input.

Flanketype	værdi	input	dataOut	Ny værdi
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	0	0
1	0	1	1	1
1	1	0	0	0
1	1	1	0	1

Flanketype ned = 0, op = 1.

Input og output bliver præsenteret af OFF og ON. Hvor ON = 1.

Flankedetektor leverer en ON impuls, som svarer til tidsintervallet mellem to kald af dataOut.

## Opsætning

I bibliotek skal der erklæres globalt:

- Enum-liste med flanketype: EDGEDOWN, EDGEUP.

I hovedprogram setup() skal der opsættes:

- <navn på flankedetektor>.begin(...)

## Klasse

Klassens navn: t\_EdgeDetector

## Medlemmer

Flankedetektors medlemmer.

Navn	Type	Egenskab	Beskrivelse
edgeType	EdgeTypes	enum	Flanketype

## Metoder

Flankedetektors metoder.

Navn	Argumentliste	Returnerer	Beskrivelse
Constructor	()		Default sætter constructor værdi = OFF og flanketype = ned
begin	(startValue: bool, edgeType: EdgeTypes)		Initialiserer flankedetektor
dataOut	(value: bool)	bool	Udlæser resultat af input

# Register

22. januar 2023 22:05

Register modtager og gemmer 1 bit til senere brug.

Brugeren af register sender reset, når register skal gøres klar til at modtage nye data.

Register kan for eksempel bruges sammen med en betjeningshandling, som først skal bringes til udførelse senere. Styringen kan være i en tilstand, som ikke skal udføre betjeningen her og nu.

## Funktion



Input og output bliver præsenteret af OFF og ON. Hvor ON = 1.

## Opsætning

I hovedprogram `setup()` skal der opsættes:

- `<navn på register>.begin(defaultValue)`

## Klasse

Klassens navn: `t_Register`

## Medlemmer

Register arver sine medlemmer.

## Metoder

Registers metoder.

Navn	Argumentliste	Returnerer	Beskrivelse
Constructor	<code>()</code>		Default sætter constructor værdi = OFF
<code>begin</code>	<code>(startValue: bool)</code>		Initialiserer register.
<code>dataOut</code>	<code>(value: bool)</code>	<code>bool</code>	Indlæser den værdi, der gemmes til senere. Udlæser den gemte værdi.
<code>reset</code>	<code>()</code>		Sætter værdi = OFF.

# Toggle

22. januar 2023 22:07

Toggle skifter mellem ON og OFF, hver gang den får ON indlæst.

Toggle kan for eksempel bruges sammen med en betjening med en trykknop, der er en tænd sluk funktion.

## Funktion

Hver gang toggle får nyt input, skal den genberegne output.

værdi	input	dataOut	Ny værdi
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

Funktionen er XOR.

Input og output bliver præsenteret af OFF og ON. Hvor ON = 1.

## Opsætning

I hovedprogram setup() skal der opsættes:

- <navn på toggle>.begin(defaultValue)

## Klasse

Klassens navn: t\_Toggle

## Medlemmer

Toggle arver sine medlemmer.

## Metoder

Toggles metoder.

Navn	Argumentliste	Returnerer	Beskrivelse
Constructor	()		Default sætter constructor værdi = OFF
begin	(startValue: bool)		Initialiserer toggle
dataOut	(value: bool)	bool	Udlæser resultat af input