

Styreenheder

1. februar 2023 14:26

Styreenheder simulerer elektronikkomponenter der leverer output til: LED, lydgivere, relæspoler som eksempler.

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: JBCtrlUnits.h.

Funktion

En styreenhed leverer output til port i en output-driver.

En styreenhed konfigureres til den aktuelle anvendelse. For eksempel:

- Blink af en LED.
- Tænd og sluk af en lydgiver.

En styreenhed har som funktion.

- En styreenhed får tilknyttet en driver og en port ved opsætning.
- En styreenhed bruger polling for eksempel til blinkende LED.
- En styreenhed modtager ny tilstand.
- I software er der brugt navngivning: Indgangens tilstand: ON eller OFF.

Opsætning

I hovedprogram skal der erklæres globalt:

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

I hovedprogram setup() skal der opsættes:

- <navn på styreenhed>.begin(...)

Grænseflade

Grænsefladen giver en ensartet tilkobling af styreenhed.

For grænsefladens design se softwarepattern facade.

Klasse

Klassens navn: t_CtrlUnit

Medlemmer

Grænsefladens medlemmer.

Navn	Type	Egenskab	Beskrivelse
driver	*OutputDriver		Pointer til output drivere
portNo	unsigned int		Output portens nummer
state	byte		Indre tilstand.

Metoder

Grænsefladens metoder.

Navn	Argumentliste	Returnerer	Beskrivelse
setPort	(driver: *OutputDriver, portNo: unsigned int)		Outputdriver initialiseres.

doClockCycle	()		Modtager polling til interne tidsstyrede funktioner.
to	(state: byte)		Modtager styreenheds næste tilstand.

Tænd sluk enhed

1. februar 2023 14:26

I en række applikationer er der behov for en simpel udgang, der blot er tænd og sluk. Brug af denne softwarekomponent gør det enkelt at tilslutte for eksempel LED eller lydgivere.

Funktion

Styreenhed sætter simpelt porten på HIGH eller LOW.

Opsætning og brug i komponent

En komponent, der skal bruge en styreenhed skal opsætte:

- Pointer til <navn på tænd sluk enhed>.

Klasse

Klassens navn: t_OnOffOut

Medlemmer

En tænd sluk enhed arver medlemmer.

Metoder

En tænd sluk enheds metoder.

Navn	Argumentliste	Returnerer	Beskrivelse
begin	(driver: *OutputDriver, portNo: unsigned int, state: byte)		Styreenhed initialiseres. State og udgangen sættes default OFF.
to	(state: byte)		Modtager ny tilstand: ON eller OFF.

Med blink

1. februar 2023 14:27

I en række applikationer er der behov for en simpel udgang, der blot kan tænde og slukke for blink. Brug af denne softwarekomponent gør det enkelt at tilslutte for eksempel LED eller lyd giver.

Funktion

Styreenhed kan tænde eller slukke for blink. Blink sætter simpelt porten periodisk på HIGH eller LOW. Styreenhed med blink er fast koblet til masterblinker.

Avancerede blink funktioner med multivibratorer er ikke indbygget. Kommer der behov for avanceret blink, skal der programmeres en ny softwarekomponent.

Opsætning og brug i komponent

En komponent, der skal bruge en styreenhed skal opsætte:

- Pointer til <navn på med blink enhed>.

Klasse

Klassens navn: `t_WithBlinkOut`

Medlemmer

En med blink enhed arver medlemmer.

Metoder

En med blink enhed metoder.

Navn	Argumentliste	Returnerer	Beskrivelse
<code>begin</code>	<code>(driver: *OutputDriver, portNo: unsigned int, state: byte)</code>		Styreenhed initialiseres. State og udgangen sættes default OFF.
<code>doClockCycle</code>	<code>()</code>		Modtager polling til blink funktion.
<code>to</code>	<code>(state: byte)</code>		Modtager ny tilstand for blink: ON eller OFF.