

# Input drivere

25. april 2023 20:38

Input-drivere konfigurerer Arduino. Der er en grænseflade, der giver mulighed for at indlæse input enten fra parallelle porte eller serielt.

Bibliotek hedder: JBIInputDriver.h.

## Funktion

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 port kan have indbygget en venteperiode for kontaktprel. Så tilstand først meldes når kontakten er stabil.

En softwarekomponent skal bruge metoden read(portNo). En softwarekomponent skal referere til enten en samling parallelle eller serielle input porte. Det skal være muligt at udskifte drivere uden behov for at opdatere En softwarekomponent.

Der er behov for 1 samling af porte per protokol. 1 for parallel, 3 for serielle, 1 for analog. Hver samling får deres egen nummerserie for porte. Porten bliver unik, fordi komponenten der skal bruge porten har en pointer til samlingen, så portens adresse = <objekt adresse>.portNo.

## Opsætning

I bibliotek skal der erklæres globalt:

- Enum: NOPEN, NCLOSED.
- Enum: INTERN\_PULLUP, EXTERN\_PULLUP.
- Enum: BOUNCE\_FILTER, NO\_BOUNCE\_FILTER.

INTERN\_PULLUP og EXTERN\_PULLUP er kun relevante hvis porten bliver indstillet til NCLOSED.

## Grænseflade

Grænsefladen giver en ensartet tilkobling af input porte.

For grænsefladens design se softwarepattern facade.

## Klasse

Klassens navn: t\_InputDriver

## Medlemmer

Grænsefladen har ingen medlemmer.

## Metoder

Grænsefladens metoder.

Navn	Argumentliste	Returnerer	Beskrivelse
doClockCycle	()		Bruges af timere
read	(portNo: unsigned int, value: *int)	bool	Udlæser resultat af digitalt input. Pointer til value bruges til analoge porte.

# Parallele porte - digitale

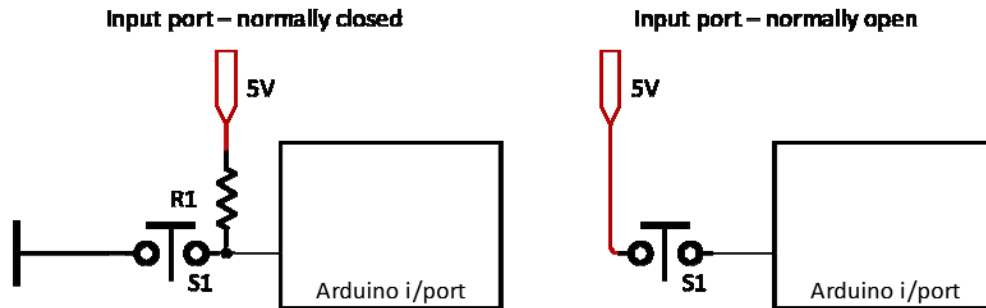
25. april 2023 17:34

Digitale input-drivere konfigurerer Arduino digitale porte med parallel kommunikation.

Driverens opgave er at konfigurere Arduino porte.

## Input portens funktion

En Arduino input port kan være indstillet til at være høj eller lav. Oversat til engelsk "Normally Closed" eller "Normally Open".



Modstanden der sætter indgangen høj, kan enten være monteret udvendigt eller indstillet i Arduino. Bliver modstanden monteret udvendigt, skal Arduino indstilles i pinmode "INPUT".

Driver afspejler funktionen:

NO/NC	Kontakt	Input	Driver
0	0	0	LOW
0	1	1	HIGH
1	0	1	HIGH
1	1	0	LOW

NO = 0, NC = 1. Kontakt passiv = 0, kontakt aktiv = 1. I software bliver 0 og 1 udtrykt ved navngivne konstanter: LOW og HIGH.

En omskifter kan stå permanent i én stilling, hvilket dog blot betyder at input i Arduino også bliver stående på én værdi.

Den praktiske anvendelse af en input port bestemmer om NC eller NO skal bruges. NO kan vælges, hvis kravet er at porten går aktivt høj. NC kan vælges for at negere input. NC kan også vælges som en sikkerhedsfunktion, hvor farligste stilling er LOW, således at en teknisk fejl melder farligste stilling.

## Input portens design

Hver port bliver præsenteret af: Pin, NOPEN/NCLOSED, INTERN\_PULLUP/EXTERN\_PULLUP, BOUNCE/NO\_BOUNCE, BounceTimeOpen, BounceTimeClose, Value, Default Value (når porten er passiv).

Tider for filter til kontaktprel:

- BounceTimeOpen: En kontakt åbnes og bryder strømmen. Tiden er default 100msek.
- BounceTimeClose: En kontakt lukkes og slutter strøm. Tiden er default 30msek.

## Opsætning

I hovedprogram skal der erklæres globalt:

- MaxNoInParrPorts: unsigned int const=nr: Antal parallelle porte. Navn på konstant er obligatorisk.
- Enum-liste med portenes navne.
- Enum-liste med pin.
- t\_DigitalParrInDrv <navn på samling input-drivere>

Yderligere konstanter kan blive sat op, hvis tider for kontaktprel også skal indstilles.

De fleste applikationer bruger parallelle input porte. Er der ikke behov for disse porte, skal antal stadig erklæres men med antal = 0.

I hovedprogram setup() skal der opsættes:

- <navn på samling input-driver>.setPort(...)

### Opsætning og brug i komponent

En komponent, der skal bruge en port skal opsætte:

- Pointer til <navn på samling input-driver>.
- Portens navn.

Komponenten bruger read metoden til at læse fra porten.

### Samlingens klasse

Klassens navn: t\_DigitalParrInDrv

### Samlingens medlemmer

Samlingen af input porte har følgende medlemmer.

Navn	Type	Egenskab	Beskrivelse
ports	[MaxNoInParrPorts]: DigitalParrInPort		Vektor med input port konfiguration
isSetup	[MaxNoInParrPorts]: bool		Holder styr på om port er initialiseret

### Samlingens metoder

Samlingen af input porte har følgende metoder.

Navn	Argumentliste	Returnerer	Beskrivelse
Constructor	()		Initialiserer alle porte og sætter dem til ikke konfigureret
setPort	(portNo: unsigned int, pin: byte, ContacType: byte, PullupType: byte, BounceType: byte)		Mapper portNo, opkobler Arduino og konfigurerer indgangen
setPort	(portNo: unsigned int, pin: byte, ContacType: byte, PullupType: byte, BounceType: byte, bounceTimeOpen: unsigned int, bounceTimeClose: unsigned int)		Mapper portNo, opkobler Arduino og konfigurerer indgangen. Desuden bliver tider for kontaktprel sat op.
doClockCycle	()		Læser input. Udfører filter for kontaktprel hvis det skal bruges.
read	(portNo: unsigned int, value: *int)	bool	Leverer driverens nuværende værdi: HIGH eller LOW. Pointer til value=nullptr og bruges ikke.

### Input portens klasse

Klassens navn: t\_DigitalParrInPort

### Input portens medlemmer

Hver input port har følgende medlemmer.

Navn	Type	Egenskab	Beskrivelse
pin	byte		Arduino pin
value	bool		Portens værdi
defaultValue	bool		Portens værdi når den er passiv
Seqs	STABLE, BOUNCE, NO_BOUNCE	enum	Portens sekvenser
seq	byte		Porten går igennem sekvens.

			Hvis ej kontaktprefilter så=NOBOUNCE
bounceWait	SimpleTimer		Holder styr på tid i filter for kontaktprel
bounceTimeClose	unsigned int	=30	Tid i msek. Se forklaring herover.
bounceTimeOpen	unsigned int	=100	Tid i msek. Se forklaring herover.

### Input portens metoder

Hver input port har følgende metoder.

Navn	Argumentliste	Returnerer	Beskrivelse
setPort	(pin: byte, ContacType: byte, PullupType: byte, BounceType: byte)		Opkobler Arduino og konfigurerer indgangen
setPort	(pin: byte, ContacType: byte, PullupType: byte, BounceType: byte, bounceTimeOpen: unsigned int, bounceTimeClose: unsigned int)		Opkobler Arduino og konfigurerer indgangen. Desuden bliver tider for kontaktprel sat op.
doClockCycle	()		Læser input. Udfører filter for kontaktprel hvis det skal bruges.
read	(int, value: *int)	bool	Leverer driverens nuværende værdi: HIGH eller LOW. Pointer til value=nullptr og bruges ikke.

# Serielle porte - digitale

16. januar 2023 16:21

Ligger klar til fremtidig brug.

På skitse niveau ses der behov for en samling porte per protokol: I2C, SPI, UART, kablet og trådløs Ethernet.

Samlingen får en klasse der arver fra grænsefladen.

Vær obs på, at serielle porte bliver koblet til en perifer enhed. Brugergænseflade på PC kræver helt anden software.

# Parallele porte - analoge

25. april 2023 11:39

Analoge input-drivere konfigurerer Arduino analoge porte med parallel kommunikation.

Driverens opgave er at konfigurere Arduino porte.

## Input portens funktion

En analog port har en 10bit analog til digital konverter og leverer en integer med værdien 0 til 1023. Driver afspejler funktionen indbygget i Arduino og leverer derfor kun en integer, som bruger af driver så må omsætte til funktion i applikationen.

Hver port bliver præsenteret af: Pin, Value.

## Opsætning

I hovedprogram skal der erklæres globalt:

- MaxNoInAnlPorts: uint const=nr: Antal analoge parallelle porte. Navn på konstant er obligatorisk.
- Enum-liste med portenes navne.
- Enum-liste med pin.
- t\_AnalogParrInDrv <navn på samling input-drivere>

Drivere med input fra analoge porte er lagt i bibliotek: JBAAnalogInDriver.h.

I hovedprogram setup() skal der opsættes:

- <navn på samling input-drivere>.setPort(...)

## Opsætning og brug i komponent

En komponent der skal bruge en port skal opsætte:

- Pointer til <navn på samling input-drivere>.
- Portens navn.

Komponenten bruger read metoden til at læse fra porten.

## Samlingens klasse

Klassens navn: t\_AnalogParrInDrv

## Samlingens medlemmer

Samlingen af input porte har følgende medlemmer.

Navn	Type	Egenskab	Beskrivelse
ports	[MaxNoInAnlPorts]: AnalogParrInPort		Vektor med input port konfiguration
isSetup	[MaxNoInAnlPorts]: bool		Holder styr på om port er initialiseret

## Samlingens metoder

Samlingen af input porte har følgende metoder.

Navn	Argumentliste	Returnerer	Beskrivelse
Constructor	()		Initialiserer alle porte og sætter dem til ikke konfigureret
setPort	(portNo: unsigned int, pin: byte)		Mapper portNo, opkobler Arduino.
doClockCycle	()		Læser input.
read	(portNo: unsigned int, value: *int)	int	Leverer driverens nuværende værdi. Den kaldende funktion har sin egen value.

## Input portens klasse

Klassens navn: t\_AnalogParrInPort

## Input portens medlemmer

Hver input port har følgende medlemmer.

Navn	Type	Egenskab	Beskrivelse
pin	byte		Arduino pin
value	int		Portens værdi

### Input portens metoder

Hver input port har følgende metoder.

Navn	Argumentliste	Returnerer	Beskrivelse
setPort	(pin: byte)		Opkobler Arduino.
doClockCycle	()		Læser input.
read	(value: *int)	int	Leverer driverens nuværende værdi. Den kaldende funktion har sin egen value.