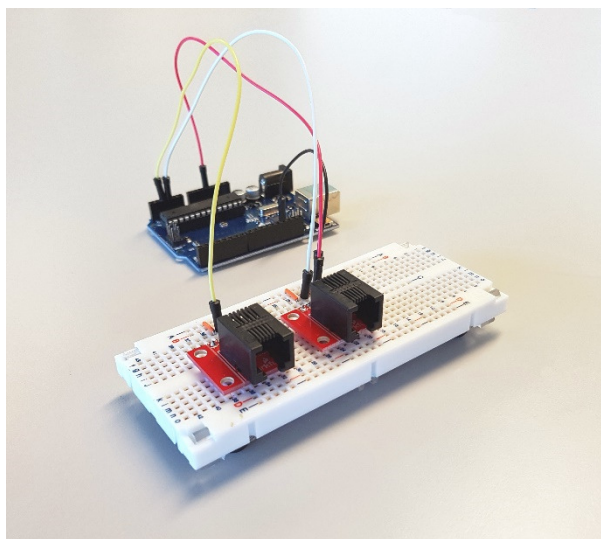


Dokument:	663010-Eksempler	Eksempler på anvendelse	
Version	2018-06-19 / HS		p. 1/4



663010 er et sæt med to moduler for tilkobling af Frederiksen sensorer til f.eks. et breadboard.

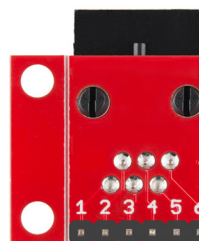
Dette muliggør loddefri forbindelse til f.eks. en Arduino.

Der medfølger to stk. krydset modularkabel til forbindelse mellem sensor og interfacemodul.

I dette dokument gives eksempler på elektriske forbindelser og tilhørende programmer til Arduino.

## Tekniske data

Ben 1	Analogt signal	Indgang til opstillingen
Ben 2	n.c. *)	(ingen forbindelse)
Ben 3	0 V	GND på Arduino
Ben 4	Digitalt signal	Indgang til opstillingen
Ben 5	5 V ud	5 V på Arduino
Ben 6	n.c. *)	(ingen forbindelse)



\*) Ben 2 og 6 anvendes ikke i denne sammenhæng. De må ikke forbindes, da de evt. anvendes internt mellem forskellige Frederiksen-apparater.

De to typer signal (digitalt eller analogt – ben 4 eller ben 1) afhænger af den tilsluttede sensor.

Sensorer med modularstik forbindes med et **krydset** kabel (1-6, 2-5 osv.) varenummer 197571. (En manual for den tilsluttede sensor vil derfor vise benenes numre i omvendt rækkefølge.) To kabler medfølger.

**Alle angivelser af ben-numre herunder henviser til den nummererede stift-række på 663010-modulet.**

## 1 – Fotoceller og lignende

### 197570 SpeedGate

Digitalt signal. Forbind ben **3, 4** og **5**.

Vælg funktionen *Dark Time (X)* på SpeedGaten. Sæt kablet i *Chain OUT*.

Det digitale signal (ben 4) er 0 V, når lyset passerer, og 5 V, når lysstråle *X* (den forreste) brydes.

SpeedGaten strømforsynes af batterier, men der skal stadig leveres 5 V på ben 5 for at aktivere signalerne fra SpeedGaten.

### 197520 Pasco fotocelle ME-9498 (indgår i yderligere Pasco-varenumre)

Digitalt signal. Forbind ben **3, 4** og **5**.

Fotocellen afgiver 5 V på ben 4, når lyset kan passere og 0 V, når lysstrålen brydes.

Bruger 5 V forsyningsspændingen (ben 5).

(Det ses, at de to ovennævnte fotoceller reagerer lige præcis omvendt, når lysstrålen brydes.)

## Program til Arduino

Dette program vil få Arduinos indbyggede lysdiode til at lyse, når lysstrålen på SpeedGate brydes. Alternativt kan en Pasco fotocelle anvendes, blot skal konstanten `invert` sættes til `true`.

```
const byte pinInput = 7;      // Forbind Arduino-pin 7 til interfacemodul ben 4
const byte pinLed = 13;      // Indbygget lysdiode
const bool invert=false;     // Ved brug af Pasco photogate: Sæt invert=true

void setup() {
  pinMode(pinInput, INPUT);  // Input fra fotocelle
  pinMode(pinLed, OUTPUT);   // Output til lysdiode
}

void loop() {
  digitalWrite(pinLed, digitalRead(pinInput) != invert);
}
```

## 2 – Afbrydere, fritfaldsudstyr osv.

Der findes en del forskelligt udstyr, som principielt blot fungerer som en afbryder. For eksempel vores 198010 Frit falds-apparat (både startdelen og faldpladen).

- Apparat, som med sikkerhed udgør en passiv afbryder, kan forbindes til en Arduino via **197572 Kabel, sikkerhedsstik til modular 1,5 m**

Der er andet udstyr, som også ligner en afbryder, men hvor der er eksterne spændinger involveret. For eksempel vores 198510 Afbryderboks, som bl.a. anvendes sammen med 195210 Affyringsmekanisme til luftpudeskinnen. Denne boks skal tilsluttes en ekstern strømforsyning, og dens *Timer Signal* er en spænding, som nemt kan overstige det, som en Arduino kan tåle.

- Apparat, som muligvis afgiver spændinger højere end 5 V, må **kun** forbindes til en Arduino via en **195212 Signalbegrænsner**

Der er intet i vejen for konsekvent at anvende en signalbegrænsner. Det simple kabel er dog billigere. I hvert fald på kort sigt...

### 197572 Kabel, sikkerhedsstik til modular 1,5 m

Digitalt signal. Forbind ben **3** og **4**. (Bruger ikke 5 V forsyningen.)

Overfører indgangssignalet som digitalt signal, direkte – uden beskyttelse. Anbefales kun i forbindelse med simple tænd-sluk afbrydere, herunder Frederiksens 198010 Frit fald-apparat. Indgangen skal konfigureres med pull-up, se eksempelprogrammet.

### 195212 Signalbegrænsner

Digitalt signal. Forbind ben **3** og **4**. (Bruger ikke 5 V forsyningen.)

Anbefales kraftigt til eksperimenter med opstillinger, hvor man ikke med garanti holder sig mellem 0 og 5 V. Overfører indgangssignalet som digitalt signal, men beskytter Arduinoen mod overspænding og forkert polaritet indenfor området 0 til  $\pm 24$  V. Indgangen skal konfigureres med pull-up, se eksempelprogrammet.

## Program til Arduino

Dette program vil få Arduinos indbyggede lysdiode til at lyse, når den digitale indgang forbindes til 0 V.

```
const byte pinInput = 7;      // Forbind Arduino-pin 7 til interfacemodul ben 4
const byte pinLed = 13;      // Indbygget lysdiode

void setup() {
  pinMode(pinSwitch, INPUT_PULLUP); // Input, brug Arduinos pullup modstand
  pinMode(pinLed, OUTPUT);         // Output til lysdiode
}

void loop() {
  digitalWrite(pinLed, digitalRead(pinInput) == LOW );
}
```

### 3 – Geiger-Müller -rør og lignende

Disse produkter ligner hinanden meget, men detaljer som spændingsforsyning og impulsernes varighed varierer.

#### 513575 GM-sensor

#### 513520 Håndholdt GM-detektor

#### 512585 GM-adapter

Digitalt signal. Forbind ben **3**, **4** og **5**.

Afgiver en impuls på 5 V på ben 4, hver gang der registreres en ioniserende partikel. Impulserne er 100 til 500  $\mu$ s lange.

Bruger 5 V forsyningsspændingen (ben 5).

#### 513610 GM-tæller

Digitalt signal. Forbind ben **3** og **4**. (Bruger ikke 5 V forsyningen.)

Afgiver en impuls på 5 V på ben 4, hver gang der registreres en ioniserende partikel. Impulserne er 45 til 50  $\mu$ s lange.

#### 513800 Koincidensboks

Digitalt signal. Forbind ben **3** og **4**. (Bruger ikke 5 V forsyningen.)

Afgiver en impuls på 5 V på ben 4, hver gang der registreres en koincidenshændelse. Impulserne er 45 til 50  $\mu$ s lange. Det kalibrerede interne koincidens-interval er kun på 1,00  $\mu$ s – udgangsimpulserne er ”strukket”.

#### Program til Arduino

Dette program vil få Arduinos indbyggede lysdiode til at give et kort glimt, hver gang der registreres en impuls.

```
const byte pinInput = 7;           // Forbind Arduino-pin 7 til interfacemodul ben 4
const byte pinLed = 13;           // Indbygget lysdiode
unsigned long last;               // Tid for seneste impuls

void setup() {
  pinMode(pinInput, INPUT);       // Input fra GM-sensor
  pinMode(pinLed, OUTPUT);        // Output til lysdiode
}

void loop() {
  if (digitalRead(pinInput)) {    // Er der en impuls i gang? - I så fald:
    last = millis();              // Tid = nu
    digitalWrite(pinLed, HIGH);   // Tænd lysdiode
  } else {                         // - ellers ... (ingen impuls i gang):
    if (millis()-last>50)         // Er der gået 50 ms?
      digitalWrite(pinLed, LOW);  // - i så fald: Sluk igen
  }
}
```

Disse produkter afgiver signaler i form af relativt korte elektriske impulser. Ovenstående simple program har ingen problemer med at registrere pulserne, men blot lidt mere komplekse programmer risikerer at miste impulser, mens programmet har travlt til anden side. Til mere seriøse anvendelser anbefales det derfor at bruge Arduinos indbyggede hardware-tællere.

## 4 – Mikrofoner

### 248600 Mikrofon

### 251570 Sondemikrofon

Afgiver såvel et analogt som et digitalt signal. Forbind ben **3** og **5**, samt signalbenet **4** (digitalt) og/eller **1** (analogt).

Det digitale signal ligger i hvile på 5 V og falder til 0 V, når lyd-**niveauet** overstiger en vis grænse.

Det analoge signal (selve lyd-**signalet**) er AC-koblet og svinger omkring 0 V. Det bør ikke uden videre tilsluttes til en Arduino på grund af forekomsten af negative spændinger. Men monterer man en modstand på 10 kΩ mellem ben 1 og ben 5, flyttes hvileniveauet op på ca. 2,5 V, hvilket uden problemer kan forbindes til en analog indgang på Arduino.

Bruger 5 V forsyningsspændingen (ben 5).

### Program til Arduino

Til test af funktionen af det **digitale** signal kan man bemærke, at opførelsen er helt som 197520 Pasco fotocelle.

Det samme testprogram kan derfor bruges.

Til at teste funktionen som **analog**-mikrofon kan nedenstående program benyttes. Bemærk, at der ikke er tale om en decideret sampling af lyden – det kan programmet ikke følge med til. I stedet kan vi betragte målingerne som stikprøver på lydniveauet og anvende disse til at estimere spidsværdier af lydniveauet.

Der skrives til PC'ens monitor-vindue i Arduino IDE'et via USB. Det er peak-to-peak signalet, der vises, med et langsomt "henfald" til nul.

```

const byte analogPinIn = 0;          // Forbind Arduino-pin A0 til interfacemodul ben 1
const int displayTime = 150;         // Udskriv hvert 0,15 s
const int calcTime = 10;             // Genberegnet middelværdi og "henfald" hvert 0,01 s
const float tau=25.0;                // Tidskonstant for "henfald" af lydtryk

void setup() {
  Serial.begin(500000);               // Baud-rate for seriel kommunikation
}

unsigned long lastDisplay=0;          // Tid for seneste udskrift
unsigned long lastCalc=0;             // Tid for seneste beregning
float x;                              // Seneste analoge inputværdi [0..1023]
float xMin=500;                       // Minimumsværdi ("henfalder" mod middelværdien)
float xMax=500;                       // Maximumsværdi ("henfalder" mod middelværdien)
float xMean=500;                      // Vejet middelværdi

void loop() {
  x = analogRead(analogPinIn);        // Indlæs ny måling
  if (x>xMax) xMax=x;                  // Er det et nyt maximum?
  if (x<xMin) xMin=x;                  // Er det et nyt minimum?
  if (millis()-lastCalc>calcTime) {   // Udfør midling og "henfald"
    lastCalc=millis();
    xMean = ((tau-1.0)*xMean + x)/tau;
    xMax = ((tau-1.0)*(xMax-xMean))/tau + xMean;
    xMin = ((tau-1.0)*(xMin-xMean))/tau + xMean;
  }
  if (millis()-lastDisplay>displayTime) {
    lastDisplay=millis();
    Serial.println(xMax-xMin);        // Udskriv p-p måling
  }
}

```