

Løsningsforslag IN1080 vår 2023

Seksjon 1

Impedans

Følgende påstand er korrekt: «Impedansen blir 0 mellom N1 og N2, og $2R$ mellom N1 og N3»

Kirchhoffs spenningslov

Følgende påstand er korrekt: «KVL tar hensyn til retningen på spenningsforskjellen over hvert element»

Seriell krets

Spørsmål 1: $V_{R2} = 6 \text{ V}$

Spørsmål 2: $I_{R1} = 2 \text{ mA}$

Parallellkrets

Spørsmål 1: $R_{\text{tot}} = 375 \text{ Ohm}$

Spørsmål 2: $I_{R4} = 10 \text{ mA}$

Spørsmål 3: $I_{V2} = 13,33 \text{ mA}$

Kondensatorer – del 1

Følgende påstand er korrekt: «Reaktansen øker hvis avstanden mellom platene blir større»

Kondensatorer – del 2

Spørsmål 1: $\tau = 0,01 \text{ s}$

Spørsmål 2: Formelen for kapazitiv reaktans gir at $X_C = 159,23 \text{ Ohm}$

Spørsmål 3: Formelen for total impedans i en seriell RC-krets gir at $Z_{\text{tot}} = 10126 \text{ Ohm}$

Operasjonsforsterkere – del 1

Følgende påstand er korrekt: «uendelig stor båndbredde»

Operasjonsforsterkere – del 2

Spørsmål 1: Dette er en inverterende forsterker hvor $A = -R_f/R_1$, som gir at $R_1 = 10 \text{ kOhm}$

Spørsmål 2: Fra formelen i forrige oppgave får vi at $V_{\text{out_min}} = -50 \text{ V}$, $V_{\text{out_max}} = 50 \text{ V}$

Wheatstonebroer – del 1

Følgende påstand er korrekt: «I en balansert Wheatstonebro er det tre kjente og en ukjent motstand»

Wheatstonebroer – del 2

Spørsmål 1: Bruker formelen for spenningsdeling, som gir $V_{BD} = 4 \text{ V}$

Spørsmål 2: Bruker formelen for spenningsdeling, som gir $V_{CD} = 4 \text{ V}$

Spørsmål 3: Summerer R_1 og R_2 i grenen til venstre, som gir $I_{R1} = 20 \text{ mA}$

Spørsmål 4: Summerer resistansene R_x og R_3 i grenen til høyre, som gir $I_{RX} = 13,33 \text{ mA}$

Dioder – del 1

Følgende påstand er korrekt: «En fysisk diode har ikke null motstand når den leder»

Dioder – del 2

Spørsmål 1: Dioden leder og spenningen $V_d = 0,7 \text{ V}$

Spørsmål 2: Dioden sperrer og spenningen $V_d = -5 \text{ V}$

Spørsmål 3: Spenningsfallet over resistoren blir 4,3 volt og da blir strømmen $I_R = 4,3 \text{ mA}$

Spørsmål 4: Dioden sperrer og det går da ingen strøm gjennom R, dvs $I_R = 0 \text{ A}$

Seksjon 2

Innebygde systemer

Hva er riktig om innebygde systemer?

50 **Innebygde systemer er gjerne tilpasset bestemte formål**

4 Innebygde systemer trenger et Real-time OS.

2 Innebygde systemer har lavt strømforbruk

10 Innebygde systemer er generelle, og kan brukes til mange applikasjoner

2 Innebygde systemer bruker alltid standardiserte busser, slik som USB og Ethernet

Hva er riktig om mikrokontrollere?

2 Mikrokontrollere kan ikke brukes til sanntidsapplikasjoner

2 Mikrokontrollere må programmeres med høynivå språk

1 Mikrokontrollere er pipelinet

14 Mikrokontrollere har Von Neumann arkitektur (alle periferienheter er koblet til den samme minnebussen)

49 **Mikrokontrollere har gjerne minne og prosessor på samme chip**

Hva er riktig om statiske variabler?

En statisk variabel holdes i minnet som en global variabel,

45 **men den kan bare brukes lokalt der den ble deklart.**

3 Statiske variabler kan ikke brukes i prosedyrer

18 En statisk variabel skifter aldri verdi

0 En statisk variabel i en prosedyre får en tilfeldig verdi hver gang prosedyren kjøres

1 Lokale variabler er alltid statiske

Motorer

Hva er riktig om motorer?

- 6 Induksjonsmotorer er synkrone
- 2 Børsteløse motorer har kun én elektromagnet.
- 8 Universalmotorer har en roterende stator og gir.
- 3 Reluktansmotorer har neodyme magneter.
- 49 **Børsteløse motorer må ha en drivekrets (inverter) for å kommuteres**

Hva er riktig om steppermotorer?

- 41 **Steppermotorer er egnet for åpen-sløyfe (open loop) kontroll**
- 5 Steppermotorer brukes stort sett trinnløst.
- 1 Steppermotorer kan ikke brukes i roboter fordi stoppstrømmen er for høy
- 9 Steppermotorer er det billigste alternativet
- 12 Steppermotorer er førstevalget for applikasjoner med høy rotorhastighet, slik som droner og modellbiler

Hva er riktig i forhold til kommutering av steppermotorer?

- 11 Ved wavekommutering er alle spolene aktive/på i alle steg
- 7 En enkoder er nødvendig for å kunne avgjøre hvilken kommuteringsstrategi som bør benyttes.
- 5 Wavekommutering går i halvsteg
- 37 **Ved fullstegskommutering er alle spolene aktive/på i alle steg**
- 8 Mikrostepping har tusen småsteg for hvert fullsteg

Hva er riktig om oppkobling av steppermotorer?

- 7 En unipolar steppermotor har færrest ledninger
- 5 En unipolar motor kan ikke drives med en H-bru
- 1 En steppermotor er det naturlige valget til maskiner som for eksempel en motorsag
- 53 **En bipolar steppermotor vil trenge en drivekrets (H-bru el.) som kan drive strøm i to retninger**
- 2 En homopolar steppermotor har seksten ledninger og må drives med tristate-logikk

Drivekretser

Hva er riktig om pulsbreddemodulasjon?

- 3 Modulasjonsfrekvens og signalfrekvens må være lik
- 55 **"Duty cycle" beskriver andel tid på.**
- 2 Digital elektronikk kan ikke benyttes til pulsbreddemodulering
- 1 Amplitude, pulsbredde og periode er alltid lik.
- 7 En motor drevet med et pulsbreddemodulert signal vil stoppe mellom hver puls slik som steppermotorer stopper

Vi har en "ideell" bryter koblet direkte i serie med en "ideell" spole i en krets. Det går strøm i spolen. Strømmen gjennom spolen går gjennom bryteren. Ingen andre kretselementer er koblet mellom disse.

Hva skjer når vi skrur av bryteren?

- 2 Strømmen i spolen øker til vi får en jevn lysbue i bryteren. Lysbuen kan vare i flere minutter.
- 11 Strømmen i spolen stopper umiddelbart
- 2 Strømmen fra spolen går ut i lufta og kombinerer magnetisk i den andre enden av spolen som en sløyfe.
- 3 Ingenting
- Strømmen i spolen avtar, men stopper ikke umiddelbart. Siden strømmen ikke har noen vei å gå så blir spennin**
- 49 **bryteren.**

Kontrollsystemer

Hva er riktig om åpne og lukkede sløyfer i et kontrollsystem?

- 4 En dobbeltsløyfe er et eksempel på en åpen sløyfe med lukket tilbakekobling.
- 39 **En lukket sløyfe har tilbakekobling**
- 10 En åpen sløyfe bruker en feedback-resistor for tilbakemelding
- 8 Et kontrollsystem med lukket sløyfe har ikke menneskelig input
- 7 Bang-bang kontroll er et eksempel på en åpen sløyfe

Hva er riktig om stegrespons (step response) ?

- 0 Kritisk damping betyr at dempingen er kritisk dårlig
- 40 **Et overdempet system har ikke oversving (overshoot)**
Et overdempet system er det samme som et underdempet system, bare at et overdempet system har oversving
- 10 (undershoot)
- 13 Et underdempet system har undersving (undershoot), ikke oversving (overshoot)
- 4 Et udempet system er kritisk dempet.

Sensorer

MEMS står for Micro electro mechanical systems. MEMS sensorer lages i silisium-chiper.

Hvilken av disse sensorene lages vanligvis med MEMS teknologi?

- 6 Gitar-pickups -- typisk spole+ magnet
- 21 **Akselerometere**
- 20 Potensiometre -- typisk metalbånd med bevegelig kontaktpunkt
- 8 Kondensatormikrofoner – kapasitiv membran
- 13 Strekkklapper -- motstandstråd av metall

Busser

Hva kjennetegner oppkoblingen i et I C system?

I²C er en punkt-til-punkt forbindelse (point to point). DVS vi kan kun ha to enheter på bussen
koblingen må ha i ledninger der i er antallet grensesnitt (interfaces) mellom enhetene.
det er to klokkelinjer (en for hver vei), én datalinje og én adresselinje.
MISO signalet må være høyt mens MOSI er lavt, og de må skiftes skiftvis på stigende og fallende flanke
Linjene må ha pullup-motstander, enhetene på bussen kan bare trekke linjene lave

Hvorfor benyttes tristate i busser?

- 18 Tristate betyr egentlig at vi har tre ledninger- en for klokke, en for + og en for -. USB og SPI bruker tristate på den
- 4 Tristate bruker vi til å overføre 3 bit av gangen
- 22 **Vi unngår kortslutning ved å sette en driver i tristate (høy impedanse) når den ikke er i bruk.**
- 10 Tristate gir oss tre logiske nivåer, 0, 1 og 2, som vi kan bruke til å øke overføringshastigheten
- 14 Når vi har tristate kan to enheter snakke samtidig på samme linje/ledning.

Seksjon 3

DC-motor

Solution DC motor task

Note: the hint provided for this task mixed up the units. Instead of

Hint: 1Nm/A is equal to $9.55 \cdot 10^{-3} \text{V/krpm}$

It should have been:

Hint: 1V/krpm is equal to $9.55 \cdot 10^{-3} \text{Nm/A}$

The provided solution covers both cases (affects parts b, c, d, e, and h)

The brushed DC motor provided during the IN1080 course has a stable speed of 90RPM without load, and a stall current of 250mA, when supplied with a DC voltage of 4.5V.

a) What is the terminal resistance of the motor?

Given the supply voltage of 4.5V and the stall current of 250mA, one can use the formula sheet or use Kirchhoff's voltage law on the motor circuit to relate the givens to the terminal resistance R:

$$V_{\text{motor}} = I \cdot R + L \cdot \frac{dI}{dt} + V_{\text{emf}},$$

Assuming steady-state operation, $L \cdot \frac{dI}{dt} = 0$

At the stall current I_{stall} , the motor is not moving so $V_{\text{emf}} = 0$

The equation becomes $V_{\text{motor}} = I_{\text{stall}} \cdot R$

$$\text{Solving for R, } R = \frac{V_{\text{motor}}}{I_{\text{stall}}} = \frac{4.5\text{V}}{250\text{mA}} = \underline{\underline{18 \text{ Ohm}}}$$

b) What is the motor constant K_T ?

Given the no-load speed of 90RPM and the supply voltage of 4.5V, one can solve for K_V using the formula sheet:

$$K_V = \frac{\omega_{NL}}{V_{\text{peak}}} = \frac{90\text{RPM}}{4.5\text{V}} = 20 \text{ RPM/V}$$

Solving for K_T (with the **CORRECT** conversion factor),

$$K_T = \frac{1}{K_V} = \frac{1\text{V}}{20\text{RPM}} = \frac{1000\text{V}}{20\text{kRPM}} * \frac{9.55 \cdot 10^{-3} \text{Nm/A}}{1\text{V/kRPM}} = 0.4775 \text{Nm/A} = \underline{\underline{478 \text{ mNm/A}}}$$

Solution according to the given **INCORRECT** conversion factor:

$$K_T = \frac{1}{K_V} = \frac{1\text{V}}{20\text{RPM}} = \frac{1000\text{V}}{20\text{kRPM}} * \frac{1\text{Nm/A}}{9.55 \cdot 10^{-3} \text{V/kRPM}} = \underline{\underline{5236000 \text{ mNm/A}}}$$

c) What is the stall torque of the motor with the supplied voltage of 4.5V?

According to the formula sheet, $T = K_T \cdot I$, so $T_{\text{stall}} = K_T \cdot I_{\text{stall}} = 478 \text{mNm/A} \cdot 0.25\text{A} = \underline{\underline{119 \text{ mNm}}}$ where K_T is carried over from part b) (with the **CORRECT** conversion factor).

Solution according to the given **INCORRECT** conversion factor:

$$T_{\text{stall}} = K_T \cdot I_{\text{stall}} = 5236000 \text{mNm/A} \cdot 0.25\text{A} = \underline{\underline{1309000 \text{ mNm}}}$$

As part of a student project, a different brushed DC motor is used to open the window blinds every morning when the alarm sets off, by turning a drum that winds the blinds. For this use case, the optimal motor speed is determined to be 20RPM. The motor's terminal resistance is 20Ohm, and the motor constant is 40V/krpm.

d) What is the motor torque when driven at the speed of 20RPM, when supplied with a DC voltage of 5V?

Assuming steady state, the circuit equation is $V = I \cdot R + V_{\text{emf}} = I \cdot R + K_T \cdot \omega$

$$\text{Solving for current, } I = \frac{(V - K_T \cdot \omega)}{R} = \frac{(5V - 40V/\text{krPM} \cdot 0.02\text{krPM})}{200\text{hm}} = \frac{4.2V}{200\text{hm}} = 0.21\text{A}$$

Solving for torque with the calculated current (with the **CORRECT** conversion factor),

$$T = K_T \cdot I = 40V/\text{krPM} \cdot \frac{9.55 \cdot 10^{-3} \text{Nm/A}}{1V/\text{krPM}} \cdot 0.21\text{A} = 0.0802\text{Nm} = \underline{\underline{80.2 \text{ mNm}}}$$

Solution according to the given **INCORRECT** conversion factor:

$$T = K_T \cdot I = 40V/\text{krPM} \cdot \frac{1\text{Nm/A}}{9.55 \cdot 10^{-3} V/\text{krPM}} \cdot 0.21\text{A} = 880\text{Nm} = \underline{\underline{880000 \text{ mNm}}}$$

A torque wrench was used to determine that the torque required to turn the blinds is 100mNm.

e) What is the supply voltage required to have the motor turn the blinds at 20RPM (assuming that the motor is designed to work with this voltage)?

First, convert the units of the motor constant K_T (with the **CORRECT** conversion factor):

$$K_T = 40V/\text{krPM} \cdot \frac{9.55 \cdot 10^{-3} \text{Nm/A}}{1V/\text{krPM}} = 0.382\text{Nm/A}$$

With the given **INCORRECT** conversion factor:

$$K_T = 40V/\text{krPM} \cdot \frac{1\text{Nm/A}}{9.55 \cdot 10^{-3} V/\text{krPM}} = 4188\text{Nm/A}$$

Then solve for the current given the required torque of 0.1Nm:

$$I = \frac{T}{K_T} = \frac{0.1\text{Nm}}{0.382\text{Nm/A}} = 0.262\text{A}$$

With the given **INCORRECT** conversion factor:

$$I = \frac{T}{K_T} = \frac{0.1\text{Nm}}{4188\text{Nm/A}} = 0.0000239\text{A}$$

Finally, solve for the voltage (with the **CORRECT** conversion factor):

$$V_{\text{motor}} = I \cdot R + K_T \cdot \omega = 0.262\text{A} \cdot 20\text{Ohm} + 40V/\text{krPM} \cdot 0.02\text{krPM} = \underline{\underline{6.04V}}$$

Solution according to the given **INCORRECT** conversion factor:

$$V_{\text{motor}} = I \cdot R + K_T \cdot \omega = 0.0000239\text{A} \cdot 20\text{Ohm} + 40V/\text{krPM} \cdot 0.02\text{krPM} = \underline{\underline{0.8005V}}$$

f) According to the data sheet, what is the maximum continuous current that the driver circuit can output?

From the attached PDF, in 6. Specifications → 6.1 Absolute Maximum Ratings (p. 4), look for Continuous output current, I_o : L293D → **600mA**

The power supply provided for this project outputs a DC voltage of 10V.

g) What is the maximum current consumption of the motor at this voltage?

The maximum current consumption occurs at stall current, and from part a) we know that

$$V_{\text{motor}} = I_{\text{stall}} * R$$

$$\text{Solving for stall current, } I_{\text{stall}} = \frac{V_{\text{motor}}}{R} = \frac{10V}{200\Omega} = 0.5A = \underline{\underline{500mA}}$$

The last step of the project is to implement Pulse Width Modulation (PWM) on a microcontroller.

h) What is the minimum duty cycle required to turn the blinds at all?

From part e), we know that the required current for turning the blinds (required torque of 0.1Nm) is 0.262A (0.0000239A with the given **INCORRECT** conversion factor)

Solving for the voltage, $V = I * R = 0.262A * 200\Omega = 5.24V$
(0.000478V with the given **INCORRECT** conversion factor)

Dividing by the supply voltage of 10V (with the **CORRECT** conversion factor):
duty cycle = $5.24V / 10V * 100\% = \underline{\underline{52.4\%}}$

Solution according to the given **INCORRECT** conversion factor:
duty cycle = $0.000478V / 10V * 100\% = \underline{\underline{0.00478\%}}$

Seksjon 4

Krets med bryter -1, 2

I utgangspunktet skal vi bruke bryteren sammen med en mikrokontroller. Vi ønsker at inngangen til mikrokontrolleren skal være høy når knappen er trykket inn. Fra databladet til mikrokontrolleren har vi at inngangen leses som lav dersom spenningen er lavere enn 0,5V og høy dersom inngangen har en spenning over 2,5V.

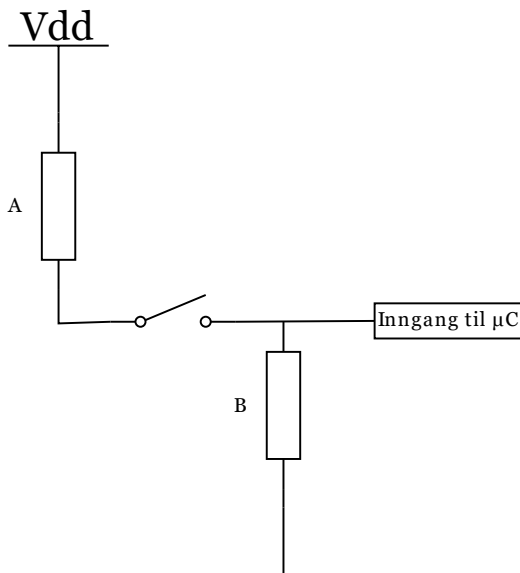
Vi regner inngangen til mikrokontrolleren som ideell, med høy inngangsimpedans.

Kretsen til bryteren kan trekke inntill 0,3mA.

Hvilken av kretsene kan brukes til formålet?

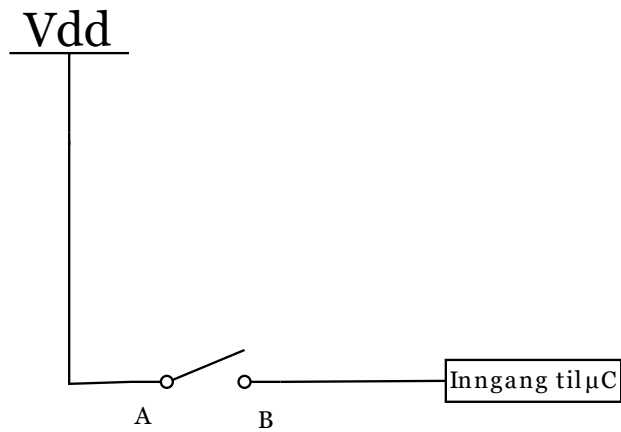
Alt som skal brukes som digital inngang til en mikrokontroller må ha en verdi som er tilstrekkelig nær Vdd eller Vss slik at mikrokontrolleren kan lese inngangen som høy eller lav. En bryter som er koblet fra gjør ingen av disse delene alene. Siden inngangen skal være høy når knappen er trykket inn, forstår vi at den må være lav når knappen er sluppet opp. For å få den lave verdien bruker vi en pulldown motstand med så høy resistans som mulig (her 10kOhm).

Når knappen kobles til, så må vi ha en pullup-motstand som har nok resistans til å begrense strømmen inn i kretsen, og som er lav nok til at spenningsdeleren vi lager når bryteren er på vil få høy nok spenning til at verdien leses som høy. Teknisk sett kunne man klart seg med motstanden som trekker ned, gitt at inngangsimpedansen er høy, men det var ikke et alternativ her, fordi vi tok høyde for at kretsen ikke alltid er ideell i utforming av alternativene.

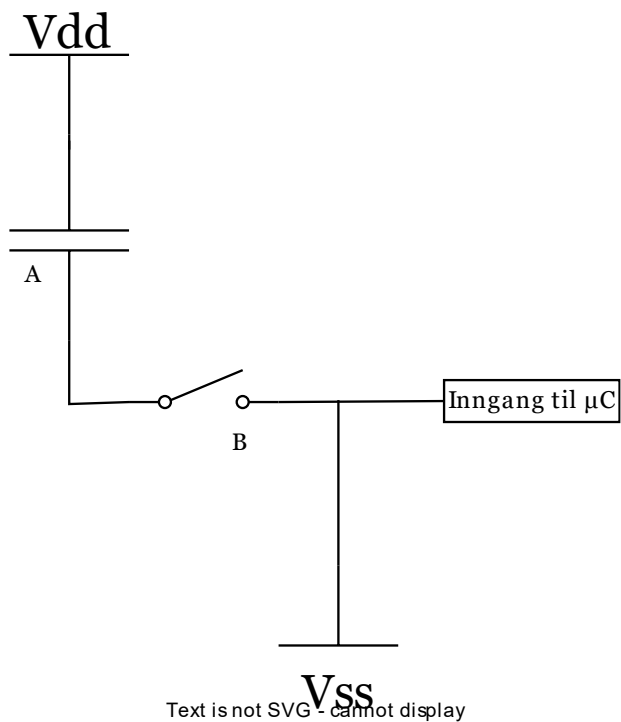


Text is not SVG - cannot display

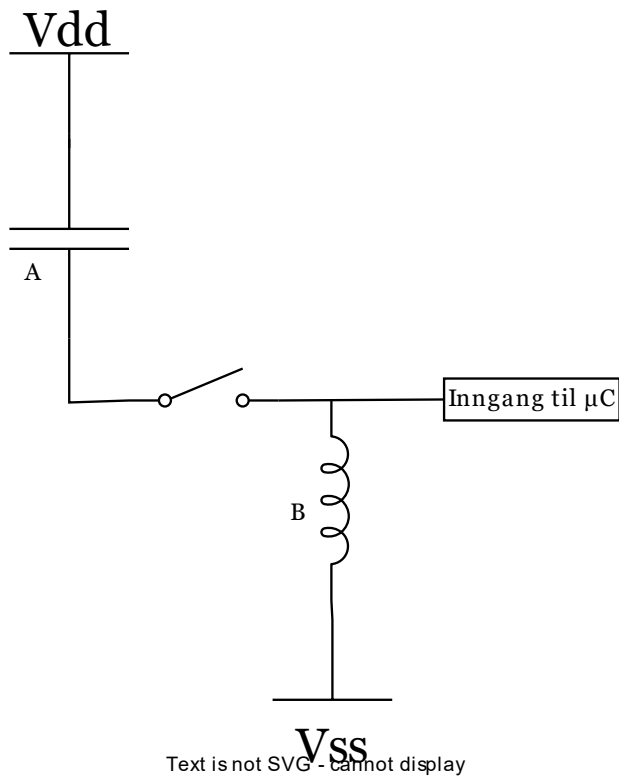
Av de øvrige løsningene så har vi



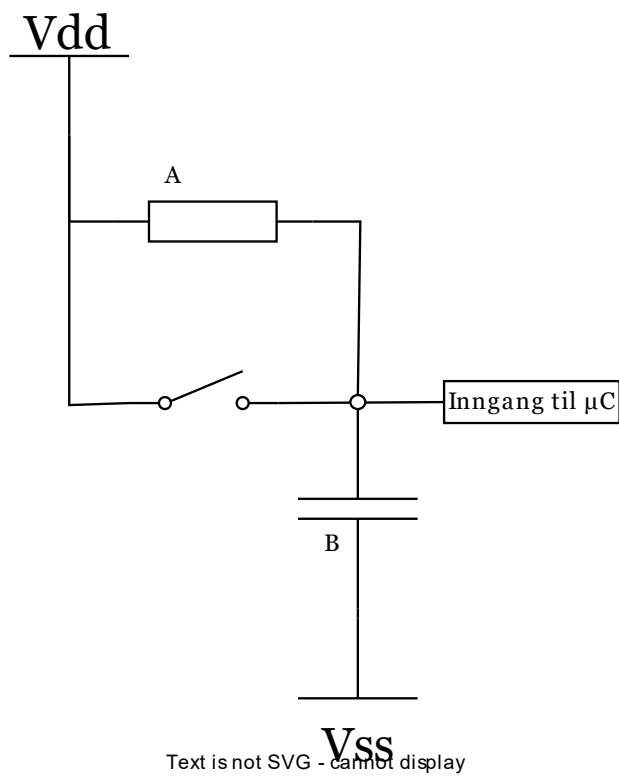
- flytende B, A koblet til V_{dd} som kan ta tilfeldige verdier, frem til bryteren er på, da vil den være høy, frem til knappen slippes og vi overlates til tilfeldigheter



- kondensator som mellom V_{dd} og bryterterminal som vil gi tilfeldig utslag til enhver tid.



- Kondensator mellom Vdd og bryterterminal, og spole som vil virke som en lavohmig pulldown motstand og alltid gi lavt resultat



- Pullupp til inngang og kondensator mellom inngang og Vss som alltid gir høyt resultat.

Hvilke verdier bør A og B fra kretsen ha?

Med utgangspunktet over er det kun to alternativer som er rimelige

A= 1kOhm

B= 10 kOhm

Hendelsesstyring

Vi ønsker å programmere en hendelsesstyring til bryteren. Hendelsesstyringen er bare en liten del av det som skal kjøres på mikrokontrolleren, så det er viktig at koden ikke oppholder andre rutiner mer enn nødvendig.

Koden skal kjøre rutinen `run_forward()` én gang når knappen trykkes inn, og `run_backward()` én gang når knappen slippes. Vi forutsetter at knappen holdes inne lenge nok til at vi kan registrere trykket.

Koden er skrevet for arduino-mikrokontrollere. Følgende kode blir brukt sammen med alternativet som skal velges:

```
const int buttonPin = 2;
void setup() {
    pinMode(buttonPin, INPUT);
}

void loop() {
    if ( buttonEvent() ){
        buttonService();
    }

    <Other checks and tasks >
}
```

Det riktige svaret tar vare på tilstand og finner endringer som kjører frem- eller tilbake-rutinen én gang.

```
boolean buttonEvent () {  
    static int lastState = 0;  
    int buttonValue = digitalRead (buttonPin);  
    if (buttonValue != lastState) {  
        lastState = buttonValue;  
        return true;  
    }  
    return false;  
}  
  
void buttonService () {  
    if (digitalRead (buttonPin) == HIGH) {  
        runForward ();  
    } else {  
        runBackward ();  
    }  
}
```

Ser vi på de gale alternativene så kan vi utelukke med følgende grunnlag:

1: While løkke blokkerer, buttonservice kjøres bare når bryteren er høy

```

boolean buttonEvent () {
    while (digitalRead (buttonPin) == HIGH) {
        buttonService ();
    }
    return false;
}

void buttonService () {
    if (digitalRead (buttonPin) == HIGH) {
        runForward ();
    }
    runBackward ();
}

```

2: Tar ikke vare på tilstand, så den vil alltid reagere om bryteren verdien er høy, men aldri om den er lav.

```

boolean buttonEvent () {
    int buttonValue = digitalRead (buttonPin);
    if (buttonValue == 1) {
        return true;
    }
    return false;
}

void buttonService () {
    if (digitalRead (buttonPin) == HIGH) {
        runForward ();
    } else {
        runBackward ();
    }
}

```

3: en rekursiv loop- ved å kjøre seg selv helt til tilstanden endres, blokkerer den effektivt all annen kjøring på mikrokontrolleren. I tillegg vil den antagelig krasje (stack overflow) siden den kjøres

rekursivt så lenge man ikke har endringer i tilstand.

```
boolean buttonEvent () {
    static int lastState = 0;
    int buttonValue = digitalRead(buttonPin);
    if (buttonValue == lastState) {
        return buttonEvent ();
    }
    lastState = buttonValue;
    return true;
}

void buttonService () {
    if (digitalRead(buttonPin) == HIGH) {
        runForward ();
    } else {
        runBackward ();
    }
}
```

4: Den siste falske kjører while løkke i service rutinen, så den blokkerer også

```
boolean buttonEvent () {
    static int lastState = 0;
    int buttonValue = digitalRead(buttonPin);
    if (buttonValue != lastState) {
        lastState = buttonValue;
        return true;
    }
    return false;
}

void buttonService () {
    while (digitalRead(buttonPin) == HIGH) {
        runForward ();
    }
    runBackward ();
}
```

Prell – 1,2

Når vi bruker bryteren, så spretter den (prell) i inntil ett millisekund. I denne omgangen skal vi håndtere prell med programmet i mikrokontrolleren. Når bryteren

IN1080-V23

25/37

trykkes eller slippes, skal det kun registreres én hendelse. Hendelsen kan registreres før bryteren har stoppet å sprette. Vi forutsetter at spretting stopper i løpet

av ett millisekund.

Vi benytter følgende kode for setup, loop og buttonService:

```
const int buttonPin = 2;
int buttonValue = 0;

void setup() {
    pinMode(buttonPin, INPUT);
}

void loop() {
    if (buttonEvent()) {
        buttonService();
    }
    <Other checks and tasks >
}

void buttonService() {
    if (buttonValue == HIGH) {
        runForward();
    } else {
        runBackward();
    }
}
```

To prosedyrer skal brukes sammen med den forhåndsdefinerte koden (over):

timed

- Skal brukes til å avgjøre når det er mer enn et millisekund siden vi registrerte forrige event.

buttonEvent

- Skal sette bryterverdien og brukes til å avgjøre om bryteren er aktivert uten at det skyldes prell.

Hvilke prosedyrer bør vi velge for timed og buttonEvent ?

Hint 1: Det kan lønne seg å se på disse oppgavene i sammenheng, selv om de kan løses individuelt.

Hint 2: Prøv å sortere ut alternativer med dårlig kode. Det er mer enn én grunn til at noe ikke virker.

Timed:

Ved å benytte koden under får vi en sammenligning med tid fra sist gang og til nå. Det er godt egnet til å se om vi har ventet lenge nok.

```
boolean timed(unsigned long lastTime){
    if (micros() - lastTime > 1000){
        return true;
    }
    return false;
}
```

Koden under kan utelukkes fordi micros() blir større enn 1000 etter ett millisekund og forblir større til vi har telt rundt, noe som tar i overkant av en time.

```
boolean timed(unsigned long eventTime){
    if (1000 > micros()){
        return true;
    }
    return false;
}
```

Koden under bruker delay som blokkerer alt annet i tiden angitt (eventtime).

```
boolean timed(unsigned long eventTime){
    delay(eventTime);
    return true;
}
```

Koden under setter verdien til en static int hver gang den kjøres (noe som er tulle), og så spør den om lasttime er større enn 1000, hvilket den vil være omtrent hele tiden (1ms per 1,2timer den ikke er det)

```
boolean timed () {  
    static unsigned long lastTime = 0;  
    lastTime = micros ();  
    if ( lastTime > 1000 ) {  
        return true;  
    }  
    return false;  
}
```

Det siste alternativet for timed vil alltid returnere false, fordi den leser av tiden to ganger rett etter hverandre, noe som ikke vil ta ett millisekund, så vi går aldri inn i løkken.

```
boolean timed () {  
    unsigned long lastTime = micros ();  
    while (micros () - lastTime > 1000) {  
        return true;  
    }  
    return false;  
}
```

For buttonEvent vil vi ha følgende alternativ, fordi den husker tilstand, og forrige gang vi skiftet tid. Den blokkerer ikke, men sjekker om vi har ventet lenge nok. Bare når vi har ventet lenge nok, så aksepterer vi en sjekk på om tilstanden har endret seg siden sist. Dette gjør det mulig å fange et trykk i det det skjer (~0 delay), mens vi etter en endring må vente ett millisekund før vi aksepterer en ny verdi. Da har prell satt seg, så vi er sikre på at ny verdi kan aksepteres.

```
boolean buttonEvent () {  
    static int lastState = 0;  
    static unsigned long lastEventTime = 0;  
    buttonValue = digitalRead (buttonPin);  
    if (timed ( lastEventTime )) {  
        if (buttonValue != lastState) {  
            lastState = buttonValue;  
            lastEventTime = micros ();  
            return true;  
        }  
    }  
    return false;  
}
```

Alternativet under kan utelukkes fordi det ikke tar vare på tidsbruken, og følgelig ikke kan håndtere prell.

```
boolean buttonEvent () {  
    static int lastState = 0;  
    buttonValue = digitalRead(buttonPin);  
    if (buttonValue != lastState) {  
        lastState = buttonValue;  
        return true;  
    }  
    return false;  
}
```

Alternativet under kan utelukkes fordi det ikke husker tiden, og det finnes ingen fornuftig timed som kan brukes sammen med den.

```
boolean buttonEvent () {  
    static int lastState = 0;  
    if (timed()) {  
        if (buttonValue != lastState) {  
            buttonValue = digitalRead(buttonPin);  
            lastState = buttonValue;  
            return true;  
        }  
    }  
    return false;  
}
```

Koden under kan utelukkes fordi 1: den akkumulerer last_event time, så det tallet øker fortere og fortere for hver gang rutinen kjøres. Hvis det tallet skal brukes til timing, så blir det bare surr.

```
boolean buttonEvent () {  
    int lastState = 0;  
    static unsigned long lastEventTime = 0;  
    lastEventTime = lastEventTime + micros();  
    if (timed( lastEventTime )) {  
        if (buttonValue != lastState) {  
            buttonValue = digitalRead (buttonPin);  
            lastState = buttonValue;  
            return true;  
        }  
    }  
    return false;  
}
```

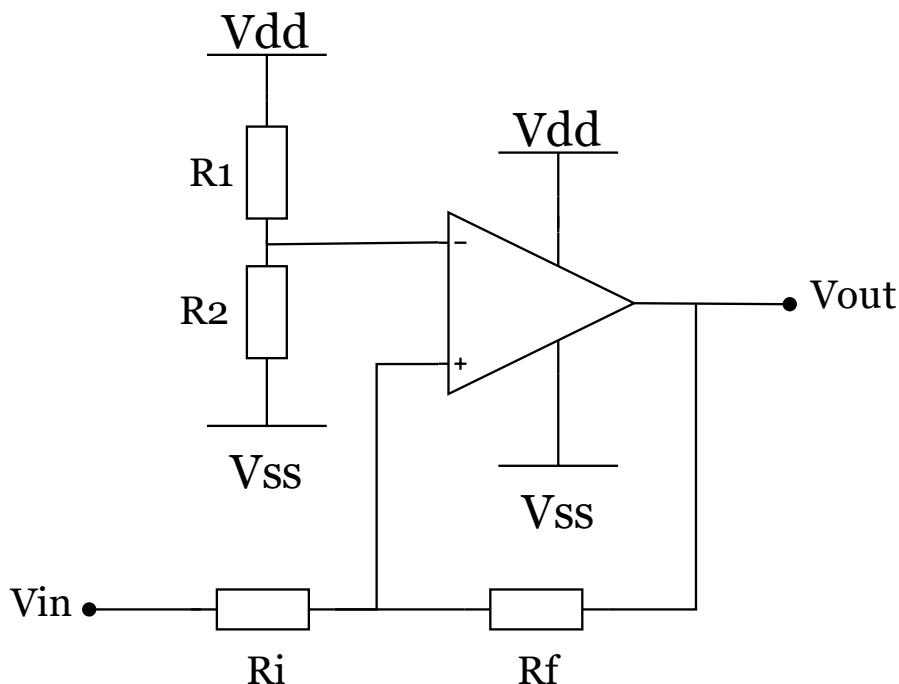
Det siste falske alternativet kan utelukkes fordi det ikke gjør noe mer enn å lese verdien av pinnen. Den husker ikke tilstand og bruker ikke timing i det hele tatt.

```
boolean buttonEvent () {  
    buttonValue = digitalRead (buttonPin);  
    if (buttonValue != digitalRead (buttonPin)) {  
        return true;  
    }  
    return false;  
}
```

Schmitt-trigger – 1, 2

I stedet for å bruke en mikrokontroller til å begrense prell, kan vi bruke en Schmitttrigger

Figuren under viser en Schmitt-trigger:



Vi lar $R_1 = R_2 = 4,9\text{k}\Omega$, og $R_i = 3,0\text{k}\Omega$. V er fremdeles $3,3\text{V}$. Vi regner operasjonsforsterkeren som ideell. Inngangen til kretsen skal maksimalt trekke $0,10\text{mA}$.

Ved hvilke inngangsspenninger skifter utgangen verdi, og hva er resistansen til R_f ?

Først så kan man finne spenningen til minusinngangen til opampen. Det er når plussinngangen har denne verdien vi vil skifte utgangsverdi (Husk ideell opamp har uendelig forsterkning, og med positiv tilbakekobling, så vil vi dermed svinge med fullt utslag når +-inngangen passerer denne spenningen).

Spenningsdeleren R_1 - R_2 gir $V_- = 3,3\text{V}/2 = 1,65\text{V}$ som inngangsverdi til minusinngangen på opampen.

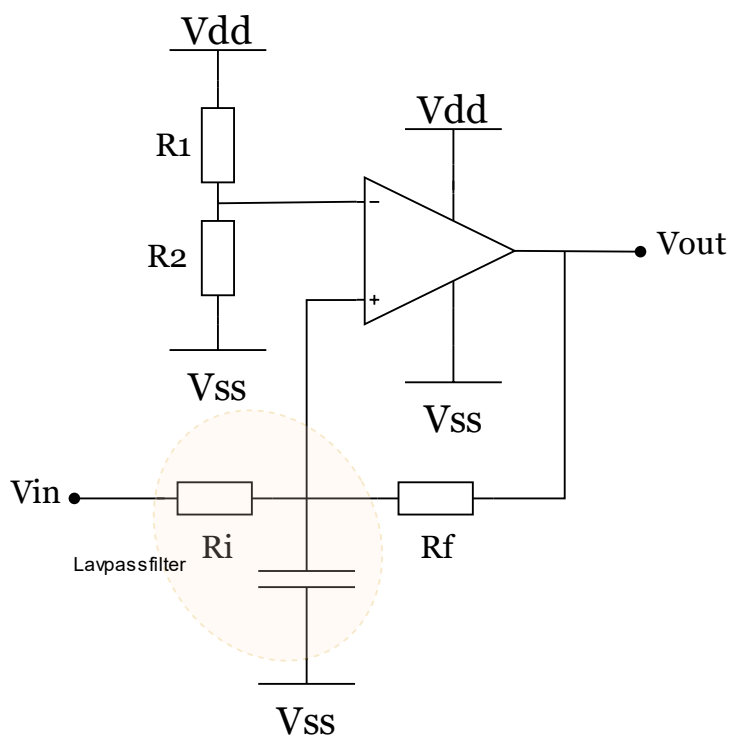
Det er oppgitt et maksimalt strømtrekk på $0,1\text{mA}$ på inngangen. Siden ingen strøm går inn gjennom en ideell opamp, får vi maksimal strøm for $I_{R_i} = I_{R_f} = 0,1\text{mA}$. Med $R_i = 3\text{k}\Omega$, så vil spenningsfallet over R_i maksimalt være $V_{R_i} = 0,1\text{mA} \cdot 3\text{k}\Omega = 0,3\text{V}$, i begge retninger. Siden vi får skifte når V_+ er lik V_- vil vi da ha en spenning på $V_+ - V_{R_i}$, når vi skifter fra høy til lav og $V_+ + V_{R_i}$, når vi skifter fra lav til høy. Setter vi inn for V_+ og V_{R_i} , får vi **overgang fra høy til lav ved $1,65\text{V} - 0,3\text{V} = 1,35\text{V}$ og fra lav til høy ved $1,65\text{V} + 0,3\text{V} = 1,95\text{V}$.**

Siden utgangen enten er 0V eller 3,3V (positiv tilbakekobling og uendelig forsterkning), så vil Spenningsfallet over R_f være 1,65V - 0V eller 1,65V - 3,3V, dvs $\pm 1,65V$. med $I_{Rf} = 0,1mA$, så må **R_f ha en resistans på $R = V/I = 1,65V/0,1mA = 16,5k\Omega$**

Merk at man kan tenke seg en situasjon der V_{in} skifter raskt, f.eks fra 3,3V til 0V mens V_{out} er på 3,3V. I det øyeblikket kan man tenke seg at vi har 3,3V fra V_{in} til V_{ut} noe som skulle diktere en større R_f for å begrense strømmen til 0,1mA. Dette ville gitt R_f på 30k Ω . Men denne situasjonen vil vare uendelig kort, fordi spenningen ved V_+ umiddelbart vil falle under terskelen for å skifte utgangen. Så det største strømtrekket vi kan få med varighet er når vi nærmer oss grenseverdien for skiftet. MAO hvis vi er uendelig nær, men ikke kommer opp til skiftet fra lav til høy med V_{out} på 0V og V_{in} på 1,95V, hvilket gir V_+ på 1,65V, og $I = 0,1mA$.

Hvilken av følgende alternativer kan brukes for å få kretsen til å filtrere prell?

- 10 En kondensator koblet mellom V_+ og enten jord (V_{ss}) eller V_{dd}
- 4 En resistor koblet mellom + og - på opampen
- 19 En kondensator koblet mellom V_{out} og jord (V_{ss})
- 3 En spole koblet i serie med utgangen til kretsen.
- 32 En kondensator koblet i serie med inngangen til kretsen.



Hvorfor bør man ha en slik komponent for å filtrere prell?

- 13 Den gir høypassfiltrering av inngangen til kretsen
- 0 Den forsinket kretsen med 0,01s
- 26 Sammen med R_f gir den høypassfiltrering av utgangen til opampen
- 25 **Sammen med R_i gir den lavpassfiltrering av inngangen til opampen**
- 3 Den gjør permabiliteten til kretsen tilstrekkelig stor