

# Labbrapport 5: Temperaturmätare

Laboranter:

**Namn1:** Henrik Goth

**Datorid:**

**Namn2:** Abdirahman Omar Ali

**Datorid:** AK0841

Datum då laborationen genomfördes: 10/12 - 19

---

Genom att skicka in labbrapporten intygar du/ni att följande regler har följts:

1. Laborationsuppgifter skall lösas självständigt av varje laborationsgrupp. Det är tillåtet att diskutera lösningar, men INTE att kopiera lösningar! Det är alltså INTE tillåtet att ge laborationsresultat eller färdiga lösningar till en annan grupp.
2. Bägge gruppmedlemmarna förväntas ta aktiv del i genomförandet av laborationen och skrivandet av rapporten. Detta inkluderar att bygga, programmera, dokumentera, testa och felsöka. Bägge gruppmedlemmarna skall kunna svara på frågor om hur laborationen genomförts och vilka resultat som erhållits.
3. Examination baseras alltid på individuella resultat

## 6 Moment 4: Drivrutin för temperatursensor - steg 2

### Uppgift 6.2.1

Analysera beräkningen, steg för steg, som utförs i **temp\_read\_celsius()**. Utgå från att **adc** har värdet 221.

Svar:

ADC\_correction kommer få värdet 21658(221\*98). Temp är ADC\_correction/1000 vilket blir 21. Det finns en if-sats som säger att om resten är större än 500 då ska talet avrundas up med ett 1.

### Uppgift 6.2.2

I **temp.c** finns en deklaration av en variabel, **adc**: (dvs. utanför funktionerna) enligt: `static volatile uint16_t adc = 221;`

Vad innebär "static"? Vad innebär "volatile"?

Svar:

Static betyder att variabeln är påtkomlig i filen den deklarerades i och volatile innebär att kompilatorn ska ej optimera där den används.

### Uppgift 6.2.3

Analysera beräkningen, steg för steg, som utförs i **temp\_read\_fahrenheit()**. Utgå från att **temp\_read\_celsius()** har värdet 27. Beräkningen utförs genom att "skala upp" och "skala ner" heltalsvärden istället för att använda flyttalsberäkning.

Svar:

Formel för konvertering från Celcius till Fahrenheit är

följande:

$$F = \frac{C * 9}{5} + 32$$

Formeln ovan säger oss att man tar Celsius graderna och multiplicerar med 9 bitar upplösning. Sedan delar man med 5 volt och adderar med 32.

**T.ex.**

$$\frac{27*9}{5} + 32 = \frac{243}{5} + 32 = 48,6 + 32 = 80,6 \text{ } ^\circ F$$

### Uppgift 6.2.4

Vad hade resultatet av beräkningen blivit om koden hade sett ut så här:

```
uint8_t temp_read_fahrenheit(void) {  
  
    return ((temp_read_celsius() * (9 / 5)) + 32);  
}
```

Antag ovan att funktionen **temp\_read\_celsius()** returnerar värdet 27.

Svar:

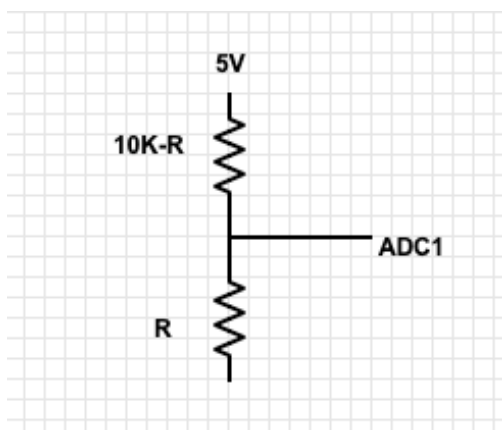
$$F = (C * (\frac{9}{5})) + 32 = (27 * (\frac{9}{5})) + 32 = (27 * 1,8) + 32 = 48,6 + 32 = 80,6 F$$

## 7 Moment 5: Drivrutin för temperatursensor - steg 3

### Uppgift 7.2.2

Rita ett kopplingsschema som föreställer spänningsdelningen och potentiometern. Bifoga detta som bild.

8



## Moment 6: Inkoppling av temperatursensor

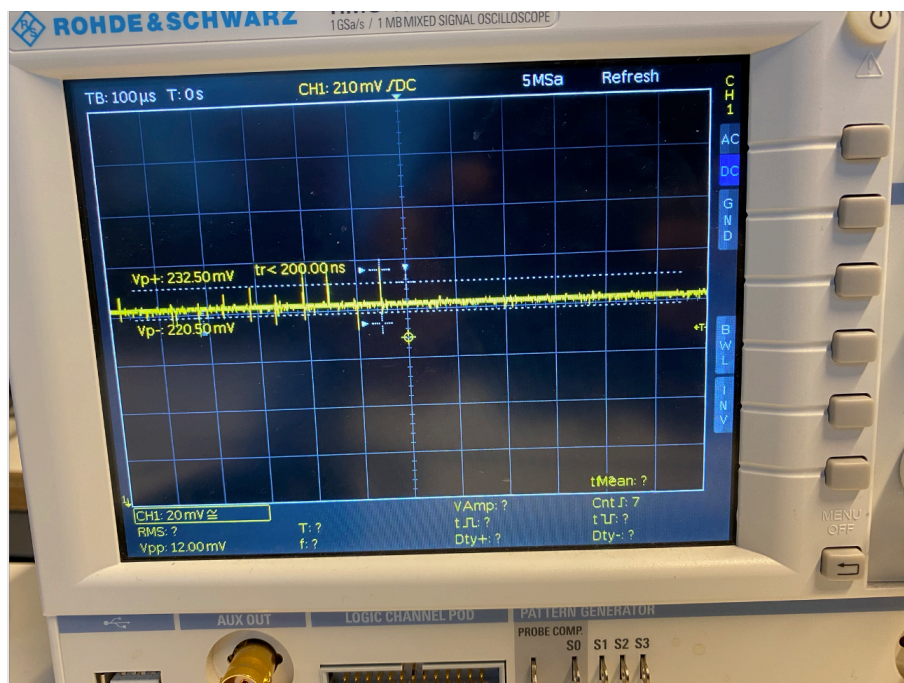
### Uppgift 8.2.2

Växla över till oscilloskopskanalens AC-läge. Ändra vertikal upplösning till någonstans mellan 10 – 50 *mV*. Finns det några störningar på signalen? Hur stora är de i så fall? Ange amplitud.

Svar: Det fanns störningar på signalen.

### Uppgift 8.2.3

Fotografera oscilloskopsbilden. Bilden ska visa oscilloskopsbilden med en vettig upplösning och stabil bild (tänk på triggingen).



### Uppgift 8.2.4

A/D-omvandlingen sker med 10-bitars upplösning. Ange antal volt per steg som denna upplösning klarar av.

Svar: 9,77mV /steg

Beräkning:  $\frac{5}{2^9} = \frac{5}{512} = 0,009765 \approx 9,77mV$

### Uppgift 8.2.5

Nu när ni känner till upplösningen och tar hänsyn till att även störningarna förstärks, kan detta påverka mätningarna? Ert svar ska redogöras noggrant och detaljerat!

Svar: När Störningar förstärks kommer det vara svårt att se hur olika kurvor utvecklas pga att mätningarna blir inte exakta.

### Uppgift 8.2.6

Om störningarna behöver hanteras, hur skulle man kunna göra detta på ett enkelt sätt? Ge förslag på minst en lösning. Ert svar behöver inte vara extremt detaljerad, men sammanfatta er på 3-5 rader. *Tips! Sök på Internet om ni inte kommer på något!*

Svar:

En lösning till att hantera störningar kan vara räkna ut genomsnittet på tillräckligt många samplingar om störningarna har samma magnitud i både positiv och negativ då kan dom ta ut varandra.

## 9 Moment 7: Test och färdigställning av system

### Uppgift 9.2.1

Nu ska ni testa så att allt fungerar enligt specifikation. Skriv ett enkelt testprotokoll där ni även anger era föregående tester med potentiometern. I övrigt bestämmer ni vad som ska testas.

Svar:

Potentiometer(%)	Spänning(V)	Grader (C)
0	0	0
14	15	1,5
16	21	2
21	25	2,5
24	30	3
25	42	3,5
26	49	4
27	49	4,5

### **Uppgift 9.5.1**

Redogör för era erfarenheter och kunskaper från denna laboration (minst en halv A4-sida):

Vad har ni lärt er?

Om ni får välja en sak, upplevde ni något som var intressant/givande?

Fanns det något som upplevdes som svårt?

Gick allting bra eller stötte ni på problem? Om allting gick bra, vad var i så fall anledningen  
detta? Om ni stötte på problem, hur löste ni i så fall dem?



## **Abdirahman:**

Laboration 5 i inbyggda system var väldigt lärofylld där vi fick en förståelse på hur man tillämpar olika tillstånd i ett program, Analog till Digital omvandling. Dessutom hur man hanterar abnrott i C programmering och att utveckla program i små steg samt hur man kan kontrollera ett värde med en potentiometer. Det vi hade mest problem med var att programmera in de olika tillstånden.

Dvs när man trycker på tangentbords knappen 1,2 eller 3 så ska den visa Celcius (knapp 1), Fahrenheit(knapp 2) och Fahrenheit och celsius(knapp 3). Som man alltid gillar att göra är att skylla på komponenterna men vi löste det med att på papper rita vårt tillstånds maskin och sedan skriva koden på papper rad för rad och desutom studera vad som händer i koden. Annat problem vi hade var LCD skärmen började bli dåligt men Magnus var snäll och gav oss en ny.

## **Henrik:**

Här blev det strax roligare än förra laborationen. Förståelsen för programmets uppbyggnad hade ökat och det var dags för tillståndsmaskinen. I detta skede blev det allt lättare för mig att förstå hur det skulle hänga ihop. Logiken bakom tillståndsmaskinen var lättare att förstå och det var en rolig utmaning att få ihop. Självklart stötte vi på vissa problem, för även om logiken är ganska intuitiv så var det ett nytt sätt att skriva en algoritm på. Att ta hänsyn till hur tillståndsmaskinen går runt och börjar om från början varje gång. Välja rätt `current_state` och hur `next_state` krävde lite tankeverksamhet. Hur skulle vi smidigast göra med de olika case och så vidare. Överlag var det en kul utmaning!