Laboration 5 – Temperaturmätaren

Laboranter:

Namn : Anna Selstam
Datorid : am3963

Namn : Malin Ramkull
Datorid : am5914

Datum då laborationen genomfördes: 2021-12-16

(Anna Selstam 2021)

Genom att skicka in labrapporten 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

Resultat

Uppgift 6.2.1 (redovisas i rapport)
Analysera beräkningen, steg för steg, som utförs i temp_read_celsius(). Utgå från att adc har värdet 221.

Svar: Först skapas en 16 bits unsigned integer vid namn adc_correction. I den sätts värdet 221 * 98 = 21658. Ytterliggare en 16 bits unsigned integer vid namn temp skapas sedan, och i den sätts 21658/1000 = 21,658. Funktionen under använder sig sedan av modulus som är en matematisk funktion som returnerar resten av en division. I detta fall testar funktionen om det blir någon rest när man dividerar adc_correction med 1000. Ifall detta restvärde är över eller lika med 500 så vill man öka variabeln temp med 1 (“round up”). Detta eftersom vår rest är över 50% av en hundradel av temp och då betyder det temp ska avrundas uppåt och inte neråt som är default.

Modulus kan beräknas med följande funktion:
21658 / 1000 =21,658
21 x 1000 = 21000
21658 - 21000 = 658

21658 modulus 1000 får alltså en rest på 658 som är större än 500. Temp ökar med 1 och returneras sedan till anropande funktion.

Uppgift 6.2.2 (redovisas i rapport)
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’ anger att variabeln endast är tillgängligt i filen. ‘Volatile’ används som modifierare som informerar kompilatorn om att variabeln delas och kan ändras utanför, exempelvis med en interrupt-rutin.

Uppgift 6.2.3 (redovisas i rapport)
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: Precis som i 6.2.1 skapar vi här två stycken 16 bits unsigned integers vid namnen convert och temp. Convert tar in det avlästa värdet i celsius och multiplicerar det med 90. Resultatet divideras sedan med 5 och till det adderas tillslut 320. I temp lagras slutresultatet dividerat med 10.
Här får vi att 27*90 = 2430. 2430/5 = 486. + 320 = 806 lagras i convert. 806 / 10 = 80,6 i temp.

Namn: Anna Selstam
Dator-id: am3963

Namn: Malin Ramkull
Dator-id: am5914

Modulus beräknas med följande funktion:
806 / 10 = 80,6
80 x 10 = 800
806 - 800 = 6

806 modulus 10, ger alltså en rest på 6 som är större eller lika med 5. Villkoret är sant så vi går in i funktionen och ökar temp med 1. Sedan returnerar vi 81,6 till anropande funktion.

Uppgift 6.2.4 (redovisas i rapport)
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);
}
```

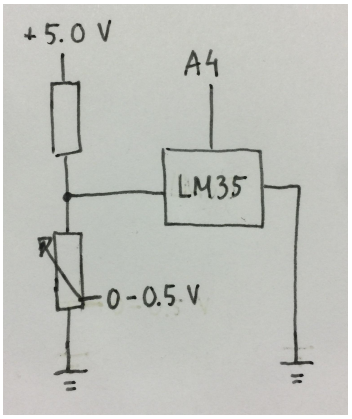
Antag ovan att funktionen temp_read_celsius() returnerar värdet 27. Tänk på vad resultatet blir när man beräknar 9/5 med heltalsaritmetik...

Svar: Convert hade fått värdet: 9/5 = 1 ⇒ 1 * 27 = 27 ⇒ 27 + 32 = 59. Testar vi sedan det med modulus 10 hade vi fått värdet 9 eftersom:
59 / 10 = 5.9
5 x 10 = 50
59 - 50 = 9

Temp = 59/10 = 5,9 + 1 från modulus-villkoret, = 6,9 = 6 hade returnerats.

Uppgift 7.2.2 (redovisas i rapport)
Rita ett kopplingsschema som föreställer spänningsdelningen och potentiometern. Bifoga detta som bild.

Svar:



Namn: Anna Selstam
Dator-id: am3963

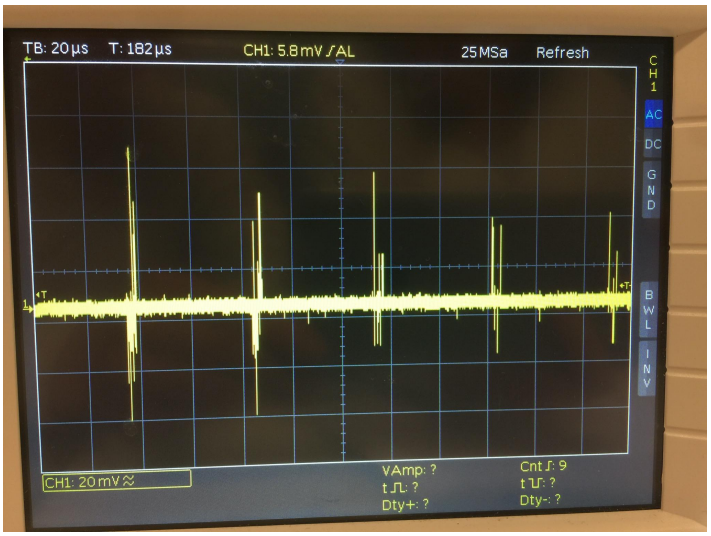
Namn: Malin Ramkull
Dator-id: am5914

Uppgift 8.2.2 (redovisas i rapport)
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: Vi upplevede en hel del störningar. Ett genomsnittsvärde på dessa störningars amplitud låg på ca 4,5 mV men detta var däremot med en upplösning på 1 mV. Med en upplösning på 20 mV kunde vi se att att det även förekom större, men färre, störningar med en amplitud på 40 mV. Se bild i nästa uppgift för förtydligande.

Uppgift 8.2.3 (redovisas i rapport)
Fotografera oscilloskopsbilden. Bilden ska visa oscilloskopsbilden med en vettig upplösning och stabil bild (tänk på triggningen).

Svar:



Uppgift 8.2.4 (redovisas i rapport)
A/D-omvandlingen sker med 10-bitars upplösning. Ange antal volt per steg som denna upplösning klarar av.

Svar:

$$Step\ size\ = \frac{Vref - Vmin}{2^n}\ (där\ Vmin\ (oftast)\ är\ 0)\ ger...$$

$$\frac{5-0}{2^{10}} = \frac{5}{1024} = 4,88\ mV\ per\ steg.$$

Namn: Anna Selstam
Dator-id: am3963

Namn: Malin Ramkull
Dator-id: am5914

Uppgift 8.2.5 (redovisas i rapport)
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: Ja, vår mätning kommer påverkas till en liten grad. Även den ideala ADC-omvandlingen inkluderar kvantiseringsfel, eftersom alla analoga ingångsvärden antas existera och de då måste de kvantiseras genom att man delar upp kontinuumet i diskreta nivåer. Dessa nivåer binds sedan ihop med det närmsta analoga värdet och presenteras som ett digitalt värde. Med detta blir alla digitala signaler exakta och kvantiseringsfelet uppstår pga att det digitala värdet inte kan presenteras som ex. 100,5 utan blir istället 100 eller 101 beroende på step size.

Den digitala output signalen som det utgörs mätningar på, ges enligt följande formel:
$d = round\ down \frac{Vin - Vmin}{step\ size}$.
Med ett step size på 4,88mV fås alltså: $d = round\ down \frac{Vin}{4{,}88*10^{-3}}$ och vi kan se att step size kommer påverka denna signal direkt.

Ju fler bitar man har i en ADC-omvandlare, desto fler blir nivåerna och desto mindre blir step size. När signalen då ska läsas av har den analoga signalen närmre till den kvantiserade signalen och kommer då få ett mindre kvantiseringsfel. Det blir med detta tydligt att ju noggrannare och mer korrekt signal vi vill ha, desto mer bitar behöver vi.

Uppgift 8.2.6 (redovisas i rapport)
Om störningarna behöver hanteras i mjukvara, 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: Offset- och Gain-errors kan motverkas med hjälp av ex. The Averaging Technique som går ut på att sampla den analoga ingången flera gånger vid bibehållandet av samma spänning, och sedan ta genomsnittet. Proverna samplas om 2 exemplar av varje eftersom det gör det mer effektivt att beräkna genomsnittet då man kan genomföra divisionen genom att högerskifta summan av de konverterade värdena. Denna typ av algoritmgenomförande, sparar CPUn både tid och minne. Nackdelen är att om en stark störning skulle uppstå, kommer detta att påverka medelvärdet.

Namn: Anna Selstam
Dator-id: am3963

Namn: Malin Ramkull
Dator-id: am5914

Uppgift 9.2.1 (redovisas i rapport)
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: Vi började med att testa så att de olika fallen för tangentvärdet stämde överens med outputen på displayen enligt tabellen nedan;

State	**key == 1**	**key == 2**	**key == 3**	**key == no_key**
SHOW_TEMP_C	C/Show C	F/Show F	CF/Show CF	C/Show C
SHOW_TEMP_C	C/Show C	F/Show F	CF/Show CF	F/Show F
SHOW_TEMP_CF	C/Show C	F/Show F	CF/Show CF	CF/Show CF

Vidare testade vi temperatursensorn genom att hetta upp fingrarna och placera dessa på sensorn och då steg värdet från dess ursprungsläge på 26°C (motsvarande 79°F) till ca 28°C - (motsvarande 82°F).

För att testa att potentiometern fungerade ökades spänningstillförseln från 0 V till 0,4 V och vi kunde konstatera att temperaturen ökade från 0°C till 43°C (motsvarande 32°F - 109°F).

Uppgift 9.5.1 (redovisas i rapport)
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?

Malin: Det var kul att denna gången få skriva main-metoden själva och applicera "status-maskin"-konceptet i vår kodstruktur. Det var även flera andra begrepp/koncept från de senare föreläsningarna som applicerades i denna labb vilket var både intressant och lärorikt.

Återigen jobbade vi med samma krets, vilket alltid är en betryggande känsla då man redan är familjär med dess struktur och funktioner. Denna gången kopplades två nya komponenter in, vilka var aningen kluriga att koppla in.

Likt den tidigare laboration var tempot under detta labbtillfälle lugnt och avslappnat. Vi stötte på en del problem, dessa diskuterades och löstes utan att ta upp allt för mycket tid. I övrigt flöt det på och det är alltid ett bra tecken när man avslutar med att redovisa under samma labbtillfälle då man påbörjade laborationen.

Anna: Denna laboration var intressant och lärorik. Det var spännande att jobba i huvudprogrammet och ge programmet en struktur och ordnad, och det flöt på relativt bra förutom i ett moment där en liten detalj hade hängt hela programmet. Som tur vad upptäcktes detta inte allt för sent m.h.a flaggor i koden.

Namn: Anna Selstam
Dator-id: am3963

Namn: Malin Ramkull
Dator-id: am5914

Det var även lärorikt att jobba vidare med kretsen på detta sätt. Koderna från tidigare labbar satt lite starkare i ryggmärgen och då förståelsen för dessa fanns till en stor grad, underlättade det vår kodskrivning i huvudprogrammet. Känslan var att man behåll en röd tråd genom hela denna laboration och att allt "knöt ihop sig" fint på slutet när kretsen och kodningen var komplett och fungerande.

Implementationen av interrupt-rutinen var den mest intressanta för mig i denna laboration. Det är fortfarande väldigt nytt, men det var lärorikt att få studera uppbyggnaden i verkligheten och att få applicera kunskapen från föreläsningarna.

Namn: Anna Selstam
Dator-id: am3963

Namn: Malin Ramkull
Dator-id: am5914