LUNDS UNIVERSITET

Lunds Tekniska Högskola

TEMPERATUR OCH VINDMÄTARE MED HÖGTALARFUNKTION

Digitala Projekt EITF 11, 18 maj 2015

Grupp 13; Morten Rosén, Henrik Boris-Möller, Christoffer Nordberg

Projekthandledare: Bertil Lindvall

Innehållsförteckning

1 Inledning

Denna rapport syftar till att dokumentera ett konstruktionsprojekt inom kursen EITF11, Digitala Projekt. Kursens mål är att ge studenterna kunskap om hur ett konstruktionsarbete går till genom att de tar fram en fungerande produktprototyp med dokumentation. I detta fall har studenterna valt att ta fram en kombinerad temperatur- och vindmätare som ger utslag i form av uppspelning av förinställda ljudklipp.

2 Kravspecifikation för prototypen

2.1 Funktionella krav

1. Prototypen ska kunna mäta temperatur.
2. Prototypen ska kunna mäta vindhastighet.
3. Prototypen ska kunna redovisa senast uppmätt data genom uppspelning av ljudklipp.
4. Prototypen ska kunna spela in ljudklipp som kan användas för redovisning av data enligt ovan.
5. Prototypen ska kunna anta ett läge där data uppmäts och ingen uppspelning sker samt ett läge där data uppmäts och redovisas via tal med ett regelbundet tidsintervall.

2.2 Kvalitetskrav

1. Prototypen ska ge värden som är korrekta inom rimlig felmarginal.
2. Prototypen ska uppdatera uppmätta värden en gång var tionde sekund.

3 Hårdvaruspecifikation

Komponenter inom parentes användes som alternativ i demonstrationssyfte.

3.1 Processor: Atmel AVR 8-bit ATmega16

Denna 8-bitars processor har ett minne på 16kb. Eftersom det behövs en lägsta samplefrekvens på 8 kHz och ett lägsta bit-djup på 8 bit för acceptabel ljudkvalitet räcker inte dessa 16kb för att innehålla tillräckligt mycket okomprimerad ljudinspelning (räkneexempel: 1 sekunder okomprimerad ljudinspelning kräver 8000 samples/sekund[1] * 8 bitar/sample[2] =64000bitar/sekund). Av denna anledning var vi tvungna att använda extern lagring av data och eftersom ISD4002 (presenterad nedan) uppfyller det kravet och dessutom har inbyggd funktion för inspelning och uppspelning av ljudklipp valde vi att komplettera ATmega16-processorn med en ISD4002.

En inbyggd A/D-omvandlare i processorn kan användas för att konvertera ett spänningsvärde till ett 10-bitars värde, vilket kommer till användning då vi ska läsa av temperaturen med LM335- temperaturmätare. Denna A/D-omvandlare kontrolleras med de båda registerna ADCSRA och ADMUX.

3.2 Temperaturmätare: LM335

Denna väldigt simpla temperaturmätare ger en outputspänning som varierar med temperaturen. Således används ATmega16's A/D-omvandlare för att konvertera denna spänning till ett värde. Självfallet måste detta värde kalibreras med en annan termometer.

3.3 Anemometer

Denna anemometer har två stycken pins. Resistansen mellan dessa pins är alltid oändligt förutom då själva propellern är i ett specifikt läge då resistansen går ner till kring 4Ω. Således kopplas denna komponents ena pin till Vcc och den andra till en input pin på processorn (PB0). Vindhastigheten kan sedan beräknas genom att räkna antal gånger PB0 är HIGH under en specifik tid.

3.4 Recording/playback device: ISD4002 (120s)

Denna komponent har ett minne för 120 sekunder ljudinspelning vilket räcker väl för vårt syfte. Den har två analog input-pinnar för mikrofonen vilka lättast används genom att koppla ena inputen till GND via en kondensator och den andra till mikrofonens output. Den input som registreras av ISD4002 är skillnaden i spänning mellan dessa två pinnar.

---

[1] (Audacity, Sample Rates, 2015)

[2] (Audacity, Bit Depth, 2014)

3.4.1 Serial Peripheral Interface

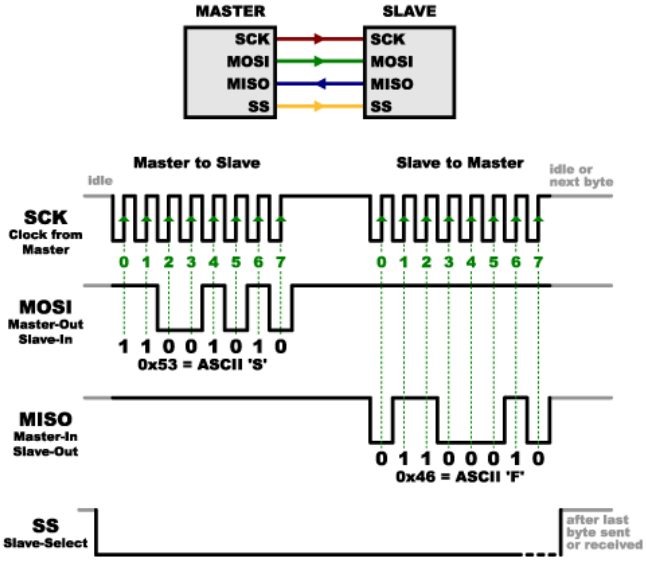
ISD4002 styrs av ATmega16-processorn med kommandon som skickas med SPI (Serial Peripheral Interface). SPI är ett standardiserat interface som används då två eller fler komponenter behöver kommunicera med varandra. Varje komponent är konfigurerad som en Master eller Slave. Mastern initierar kontakt och startar interaktionen med och kan sedan vänta på svår från Slave-komponenten. Detta sker med hjälp av fyra portar: **SCK**, **MOSI**, **MISO** och **SS**. I vårt fall är ATmega16 Master och ISD4002 Slave.

3.4.1.1 MOSI är output för Mastern och input för Slaven (Master Out Slave In). Denna port används för att skicka kommandon från Mastern till Slaven. I vårt fall med ISD4002 skickas kommandon i 16-bitarspaket.

3.4.1.2 MISO är input för Mastern och output för Slaven (Master In Slave Out) och används för att Slaven ska kunna kommunicera med Mastern. I vårt fall används denna port av Slaven till exempel för att kommunicera till ATmega16-processorn att ett ljudklipp är färdiguppspelat.

3.4.1.3 SCK bestämmer takten för interaktionens bit-rate. Denna pin är output för Mastern och input för Slaven och kan ses som en dirigents taktpinne. Denna pin behövs för att Slaven ska veta när den ska läsa av nästa bit från MOSI-pinnen och för att Slaven ska veta när den ska skicka ut data via MISO.

3.4.1.4 SS används av Mastern för att initiera kommunikation med Slaven. Då SS är LOW hos Slaven registrerar den inputs via SCK och MOSI och då SS går HIGH igen hos Slaven slutar den registrera andra inputs och börjar agera efter de kommandon Mastern skickat. SS är såklart output för Mastern och input för Slaven.



Figur 1 Tidsschema över SPI-kommunikation

De komponenter som används enbart för ISD4002 är:

3.5 5V-3.3V regulator: LM3940

Eftersom ISD4002 kräver 3,3V var denna komponent nödvändig. Även SPI inputs ska vara 3,3V (ATmega16 HIGH är 5V) och således behövdes spänningsdelare mellan processorn och ISD4002.

3.6 Mikrofon:

Denna mikrofon fungerar som ena resistorn i en spänningsdelare så att inputspänningen till ISD4002 varierar med resistansen och således varierar också med vibrationerna i luften (ljud).

3.7 Förstärkare: (Velleman MK190)/LM4860M

Den ursprungliga förstärkaren LM4860 fungerade utmärkt till en början men då vi fick oväntade problem så bestämde vi oss för att införskaffa en Velleman MK190 (4-32Ω impedans och 2*5W). Då kursen heter Digitala Projekt och inte Analoga Projekt har vi inte allt för stora skamkänslor över att vi tog denna genväg. Kopplingsschemat visar hur vi kopplade LM4860M.

3.8 Högtalare: 8Ω impedans, 10W

3.9 15V AC/DC adapter till förstärkaren: JA150-020-D

För övriga mindre komponenter hänvisas till kopplingsschemat.

4 Arbetsgång

4.1 Planering

I samverkan med handledare kom gruppen i projektets startskede fram till vilka komponenter som skulle kunna användas för att uppfylla kravspecifikationen. När dessa var bestämda ritades ett kopplingsschema med hjälp av programmet PowerLogic (se länk: Kopplingsschema). Till hjälp i detta arbete användes datablad tillhörande de olika komponenterna.

4.2 Konstruktion och testning

Konstruktionsfasen började med att koppla och löda alla komponenter enligt kopplingsschemat. Därefter testades alla komponenter i tur och ordning i Atmel Studio. Knapparna var opålitliga så kondensator och resistor implementerades. För att testa ljudinput och ljudoutput till och från ISD4002 användes ett oscilloskop. Med oscilloskopet underlättades det att felsöka hela ljudinspelnings-/uppspelningsfunktionen. Under testning av LM4860-förstärkaren (med en inledande god funktionalitet) slutade densamma efter en viss tid att skicka signal vidare till högtalaren vilket gjorde att ett alternativ behövde införskaffas. Valet föll

på en Velleman MK190-förstärkarbyggsats.

4.3 Mjukvara

All programkod för prototypens testning och bruk skrevs i språket C med hjälp av programmet Atmel Studio. Kodningen tog vid så fort konstruktionen nått en nivå där hårdvaran kunde börja testas. För testning skrevs ett program som testade varje komponent var för sig. Sedan när tillräcklig intuition om varje komponent var uppnådd skrevs den slutgiltiga programkoden. Dessutom skrevs ett program vars syfte var att spela in och lagra de olika ljudklippen. Detta program kördes bara en gång (därefter var ljudklippen permanent lagrade på ISD4002).

5 Diskussion

Den största utmaningen som gruppen ställdes inför var att förstå uppbyggnaden av varje komponent och hur de skulle samverka. Både ISD4002 och LM4860 var relativt komplicerade med många funktioner och kopplingar. Detta löstes med hjälp av att studera datablad och efter koppling prova sig fram med hjälp av oscilloskop, testmjukvara och debugging. Problematiskt också var att ISD4002 är en relativt sällan använd komponent (särskilt i kombination med ATmega16) vilket gjorde att dokumentation på internet var begränsad.

Det huvudsakliga problem som dök upp var, som tidigare nämnt, att LM4860-förstärkaren slutade ge signal till högtalaren. Eftersom de olika högtalaralternativ som gruppen hade till sin förfogan inte gick att driva utan förstärkare, och ingen backup-förstärkare fanns för omedelbar tillgång, införskaffades en enkel förstärkarbyggsats från Kjell & Company. Denna fungerade adekvat som ersättare.

Projektet var väldigt lärorikt eftersom gruppen var tvungen att själv ta reda på nästan all information kring hur de individuella komponenterna skulle användas och kommunicera med varandra via SPI. Det var intressant att prova på att styra fysiska komponenter med hjälp av programmering i C.

6 Källhänvisning

Audacity. (2014, 10 30). *Bit Depth*. Retrieved from Audacity:
http://wiki.audacityteam.org/wiki/Bit_Depth

Audacity. (2015, 4 19). *Sample Rates*. Retrieved from Audacity:
http://wiki.audacityteam.org/wiki/Sample_Rates

Stort tack till Berra och Andreas för all hjälp när vi, okunniga I-studenter, föll till föga