

Väderstation

Digitala Projekt, EITF11

Martin Olofsson, I08 & Fredrik Oher, I08

Vt 2012



The aim of this course Digital Projects was to plan, construct and program an electrical device, over the duration of eight weeks. The authors of this report have created a fully functioning weather station. This device can read the temperatures from two different locations simultaneously as well as warn the user if the temperatures override limits that the user can choose and set manually by him- or herself. This project has been quite challenging, perhaps mostly because of the authors previous lack of experience in this field. It has, nonetheless, also been very educative.

Innehåll

Inledning.....	3
Krav.....	4
Komponenter	4
AVR ATmega16.....	4
Skärm.....	4
Knappsats och encoder	5
Temperaturmätare.....	5
Konstruktion.....	5
Mjukvara.....	6
Skärmläge 1	7
Skärmläge 2	7
Skärmläge 3	8
Skärmläge 4	8
Resultat.....	9
Slutsats och Diskussion.....	9
Referenser	10
Källkod	11

Inledning

I denna kurs i Digitala Projekt har vi haft som uppgift att planera, konstruera och programmera en fungerande prototyp¹. Vi har valt att konstruera en så kallad väderstation med två separata temperatursensorer. Projektet har varit utmanande men också givande.

I denna rapport kommer vi att börja med att presentera de krav vi satte upp för vår konstruktion, som döpts till "TURE". Vi kommer sedan att gå igenom de olika komponenter som ingått i vår konstruktion, och försöka förklara vilken roll de har fyllt. Efter detta ska vi försöka förklara hur konstruktionen är uppbyggd och sedan redovisa programmet vi har skapat, och hur detta samverkar med konstruktionen. Slutligen presenterar vi resultat och sedan vår slutsats och påföljande diskussion.

Som bilaga finner ni programmeringskoden samt referenser till de datablad vi har använt oss av.

¹ Se kursmål för EITF11.

Krav

- Systemet skall läsa temperatur från två stycken analoga sensorer.
- Systemet skall i grundläget kontinuerligt visa aktuella temperaturer från sensor 1 och sensor 2.
- En användare skall kunna programmera larmgränser (övre och undre) för en av sensorerna via en knappsats.
- Systemet skall meddela användaren när alarmet för övre eller undre gräns är aktivt.
- Systemet skall kunna varna användaren när larmgränserna överskrids .
- Systemet skall kunna presentera max- och mintemperatur.
- En användare skall kunna återställa max- och mintemperaturerna.
- Systemet skall i "menyläge 2" visa mätningens max- och mintemperatur.
- Systemet skall i "menyläge 3" låta användaren skriva in larmgränser.

Initialt hade vi även tankar på funktioner kopplade till tidsangivelse. Vi insåg dock tidigt att eftersom området var så pass nytt för oss så fick ovanstående krav anses tillräckligt.

Komponenter

Följande komponenter har använts:

- En AVR ATmega16 mikroprocessor
- En fyrradig skärm av typen HD447800
- En 16-knapps knappsats
- En encoder av typen MM74C922
- Två temperaturmätare av typen LM335
- En transistor av typen BD176
- En variabel spänningsregulator
- Två kondensatorer
- Två motstånd

Dessutom användes

- en JTAG ICE mkII

AVR ATmega16

ATmega16 är en 8-bitars mikroprocessor med 16 kilobytes programmerbart flashminne. Processorn har 32 pinnar för in- och ut-kommunikation. De 32 pinnarna är uppdelade i 4 portar (A-D) med olika funktioner. Processorn har bland annat även inbyggd A/D-omvandlare, olika typer av minne samt JTAG interface för kommunikation med dator.

Skärm

Vår skärm är av märket Hitachi och av typen HD447800, även kallad "Dot Matrix Liquid Crystal Display Controller/Driver". Displayen är 4-radig med 20 teckenplatser på varje rad och mottager information av mikroprocessorn om vad som ska skrivas ut. Skärmens kontrast styrs av en reglerbar spänningsregulator. En transistor är kopplad till skärmens backlight och skärmen kan på så sätt fungera som "larm" när temperaturen över- eller understiger de inprogrammerade temperaturgränserna.

Knappsats och encoder

Vår encoder är av märket National Semiconductor Corporation och av typen MM74C922 och används för att tolka och skicka vidare information från vår knappsats till mikroprocessorn, där informationen behandlas. Encodern innehåller bland annat ett internt register som kan minnas den sista knapp som har tryckts ner, även efter att den ej är nedtryckt längre. Encodern behöver två kondensatorer för att fungera.

Temperaturmätare

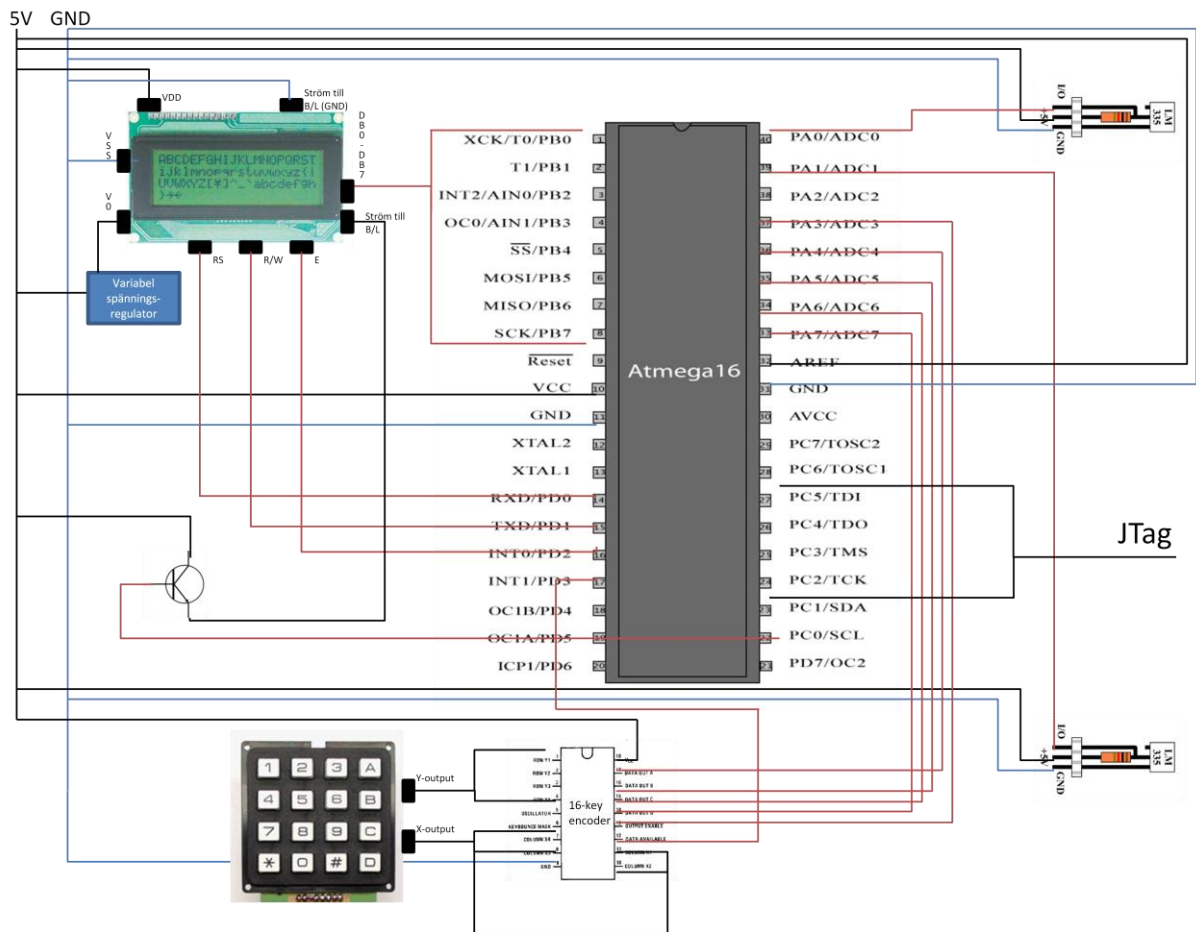
Våra temperaturmätare kommer ifrån National Semiconductor Corporation och är av typen LM335. Dessa temperaturmätare mäter i Kelvinskalen. Mätaren ger utslag med 10 mV för varje Kelvingrad. Absoluta nollpunkten (0 Kelvin) ger alltså 0 mV, 0 grader Celsius (273 Kelvin) ger 2730 mV och till exempel 20 grader Celsius (293 Kelvin) ger 2930 mV. Vårt program är konstruerat för att omvandla denna spänning till grader Celsius som sedan presenteras på konstruktionens skärm. Vi bedömde att detta var enklare än att använda en temperaturmätare som mäter i Celsiusskalen (eftersom detta skulle innebära problem vid minusgrader). Motstånd användes för att temperaturmätarna skulle fungera.

Konstruktion

Komponenterna har alla satts fast på en monteringsplatta. Via huvudsakligen virad tunn metallkabel har de sedan kopplats in till mikroprocessorn på lämpliga pinnar. Vissa pinnar på processorn är gjorda för speciella ändamål och till exempel har temperaturmätarna kopplats in till pinnar med inbyggd A/D-omvandlare. Ett annat exempel är output enable-pinnen på encodern som var tvungen att bli ihopkopplad med en interrupt-pinne på mikroprocessorn för att systemet skulle fungera.

Om mikroprocessorns portar kan man mer ingående säga att A0 och A1 tar emot temperaturmätare 1 respektive temperaturmätare 2. A3 (A2 används inte) är kopplad till output enable som skickar en utsignal till encodern. A4-A7 tar emot fyra bitars data från encodern. B0-B7 skickar data till skärmen. C0 styr om backlighten på skärmen ska vara på eller ej. C2-C5 är reserverade till JTAGen och kan således kommunicera med datorn och vårt program. D0-D2 styr displayen (det vill säga bestämmer vad som ska hända på skärmen, i jämförelse med B0-B7 som bara skickar data till skärmen).

För mer utförlig översikt, v.g. se nedanstående konstruktionsbild.



Mjukvara

Vårt program är skrivet i programmeringsspråket C, i utvecklingsmiljön AVR Studio 4.

När programmet initieras sätts alla register till följande startvärden:

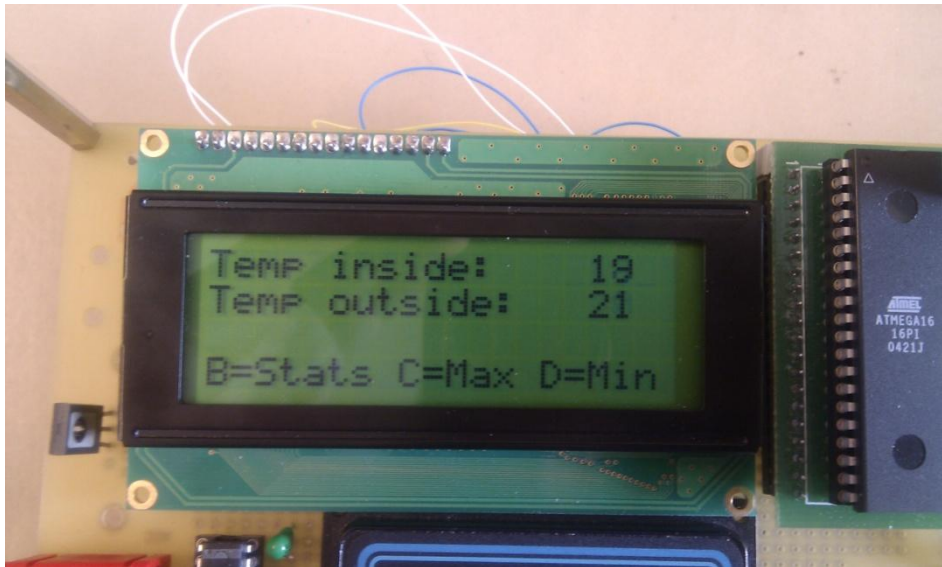
- I Data Direction Register sätts vilka portar som ska vara in och vilka som ska vara ut.
- Interruptregistret sätts till att lyssna på Port INT1 (Port D3)
- Temperaturregisterna sätts till att lyssna på temperaturmätarna på Port ADC 0 och ADC1 (Port A0 och A1).

I övrigt består programmet av två så kallade interrupt-metoder. Den första stoppar programmet när en A/D-omvandling är utförd (en ny temperaturmätning uppmäts). Den andra metoden stoppar programmet om en knapp på knappsatsen trycks ner.

Resterande delar av programmet handlar om hur data skrivs ut på skärmen och hur inmatning av till exempel larmgränser går till. För mer utförlig överblick, v.g. se källkoden i sin helhet i bilagan.

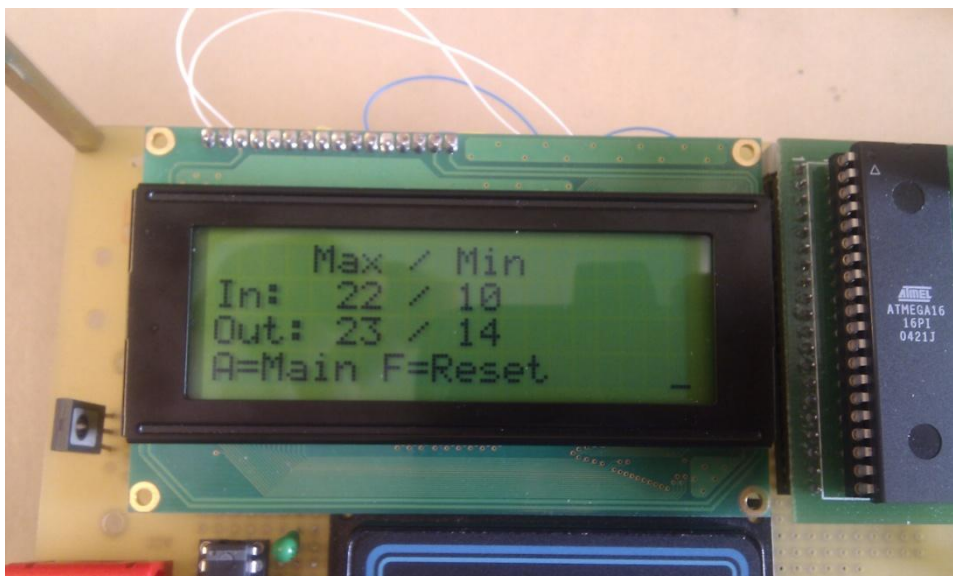
Skärmläge 1

I skärmläge 1 visas in- och yttertemperatur samtidigt. Man har även möjlighet att ta reda på mätningens högsta och lägsta värden (via knapp B) samt sätta max- och mintemperatur (via knapparna C och D).



Skärmläge 2

I skärmläge 2 kan man se mätningens max- och mintemperaturer. Man kan välja att nollställa statistiken (via knapp F) och att gå tillbaka till huvudläge (skärmläge 1, via knapp A).



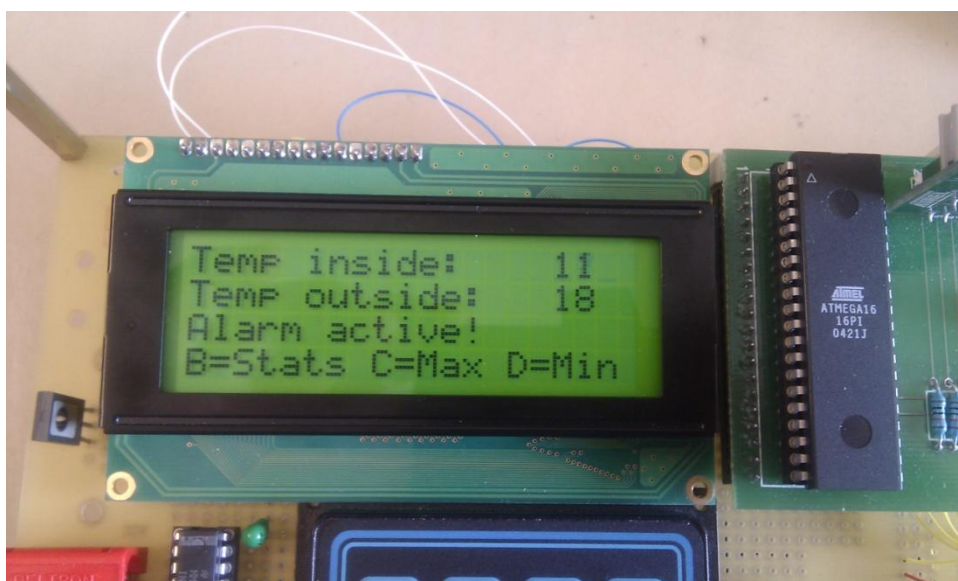
Skärmläge 3

I skärmläge 3 kan man sätta tillåten max- eller mingräns. Man kan välja minusgrader (via knapp E), att spara och gå tillbaka till huvudläge (via knapp F) och att gå tillbaka till huvudläge utan att spara (via knapp A).



Skärmläge 4

I skärmläge 4 visas texten "Alarm Active!" när min- eller maxgräns är bestämd. När gränserna överskrids kommer skärmen att lysa upp. Man kan inaktivera larmet genom att trycka på vilken sifferknapp som helst på knappsatsen.



Resultat

Med tanke på våra i princip obefintliga förkunskaper upplever vi oss nöjda med resultatet. Systemet, som döpts till "TURE", fungerar som det ska och uppfyller våra krav. Vi har under arbetets lopp haft en hel del problem med temperaturmätningen i sig. När vi har uppmätt spänningen från temperaturmätarna manuellt så har dessa bedömt korrekta, men när de behandlats i processorn och presenterats i mjukvaran har de visat andra värden. Vi kunde inte finna en felkälla till detta och fick således kalibrera datorprogrammets tolkning av temperaturgivarna för att få rimliga temperaturangivelser. Vi vet att andra grupper i år, och även tidigare år, har haft i princip identiska problem.

Slutsats och Diskussion

Generellt kan man säga att vi är stolta över vad vi har åstadkommit. Det har varit väldigt utmanande stundtals, särskilt med tanke på de tidigare nämnda i princip obefintliga förkunskaperna. Dessutom har det stundtals varit frustrerande att spendera åtskilliga timmar på att söka felkällor för att få systemet att fungera. Vi har dock utvecklats mycket av dessa svårigheter och vi upplever att vi under resan har lärt oss mycket om elektro- och datorteknik och konstruktionsprocessen, samt hur man bedriver digitala projekt.

Om man skulle utveckla vårt system ytterligare så tror vi att nästa steg är att implementera någon form av klockfunktion, för att på så sätt kunna ta fram medeltemperaturer samt olika datums min- och maxtemperaturer. Hade man velat utveckla konstruktionen ännu mer hade nästa steg varit att koppla in andra funktioner, såsom vindmätare och barometer.

Referenser

Mikroprocessor:

<http://www.atmel.com/Images/doc2466.pdf>

Temperaturmätare:

<http://www.eit.lth.se/fileadmin/eit/courses/edi021/datablad/Sensors/Im335.pdf>

Skärm:

<http://www.adafruit.com/datasheets/HD44780.pdf>

16-knapps encoder:

<http://www.elexp.com/tips/74C922.pdf>