

## Abstract

Syftet var att konstruera en väder station som håller koll på temperaturen. Huvudfunktionen var att få en grafisk visning av temperaturen över ett visst tidsintervall eftersom vi valde den grafiska displayen. Även mer triviala funktioner som aktuell temperatur samt olika beräkningar av detta så som medel temperatur och max och minvärde finns med i projektet. Projektet praktiska natur gjorde att vi försökte få alla delar att funka var och en för sig och sedan sätta ihop det till en enhet. Vi lyckades nästan uppnå vår kravspecifikation. Funktionaliteten är bra men användbarheter är mindre bra då menyerna tar lång tid att navigera igenom. Det finns endel programmerings arbete som kan förfinas ytterligare men i brist på tid så nöjer vi oss med en fungerande lösning.

## Innehållsförteckning

Abstract.....	1
1 Inledning .....	3
1.1 Kravspecifikation:.....	3
1.2 Funktionsbeskrivning:.....	3
1.3 Genomförande: .....	3
2 Hårdvaran.....	4
2.1 Komponenter.....	4
3 Mjukvara .....	5
3.1 Koden.....	6
char chargen(int in) .....	6
void init(void) .....	6
void SIG_OUTPUT_COMPARE0(void) .....	6
void busy(int chip) .....	6
void comandLCD (uint8_t com,int chip).....	6
void writeLCD(uint8_t wri,int chip).....	6
void write(char ch, int chip).....	6
void clearline(uint8_t line,int chip).....	6
void fillline(uint8_t line,int chip).....	6
void clear(int chip).....	6
void fill(int chip) .....	7
void ADConv(void) .....	7
void printTemp(uint8_t y, uint8_t x, uint8_t sid, int temp).....	7
4 Resultat .....	7
5 Förbättringsförslag.....	7
6 Slutsats .....	7
7 Referenser.....	8

## Figur- och tabellförteckning

Figure 1 Blockschemat över hårdvaran .....	4
Figure 2 Blockschemat över programmet .....	5

# 1 Inledning

## 1.1 Kravspecifikation:

Material:

Vi har valt att använda AVR Mega 16 processor, och AVR JTAG ICE för att kommunicera med datorn. I vårt projekt hade vi tänkt använda oss av en 128 X 64 pixels display för att presentera information. Tre knappar behövs för att orientera sig i menyerna. En lysdiod för att indikera om temperatur gränserna har överskridits eller underskridits. Två temperaturgivare för att mäta temperaturen både inne och ute.

## 1.2 Funktionsbeskrivning:

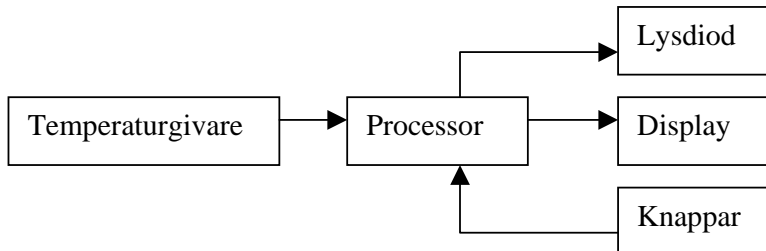
Väderstationen ska kunna visa två temperaturer kontinuerligt samt aktuell tid. Temperaturen bär uppdateras ca var 10:e sekund; dessutom lagras ett värde var 10:e minut som sparas i 24 timmar för att användas till grafen. Max och min värde för var och en av givarna sparas för det senaste dygnet. Medelvärde för den senaste timmen kan även presenteras. Med hjälp av knapparna kan man kunna orientera sig i menyerna och ställa in värden så som temperatur gränser och korrigera den aktuella tiden. Om gränserna överskrid aktiveras larmet och detta indikeras med att den röda lysdioden märkt Larm tänds.

## 1.3 Genomförande:

Vi började med att få kontakt med processorn och lyckades med det under veckan 2 samtidigt fick vi LM 35 temperatur sensorn att fungera med en från dess datablad föreslagen uppkoppling. Denna uppkoppling använde vi däremot inte i vårt slutliga projekt.

## 2 Hårdvaran

Här syns ett blockschema över vår konstruktion. Mer detaljer kan fås ur kopplingsschemat i Bilaga 1.



**Figur 1** Blockschema över hårdvaran

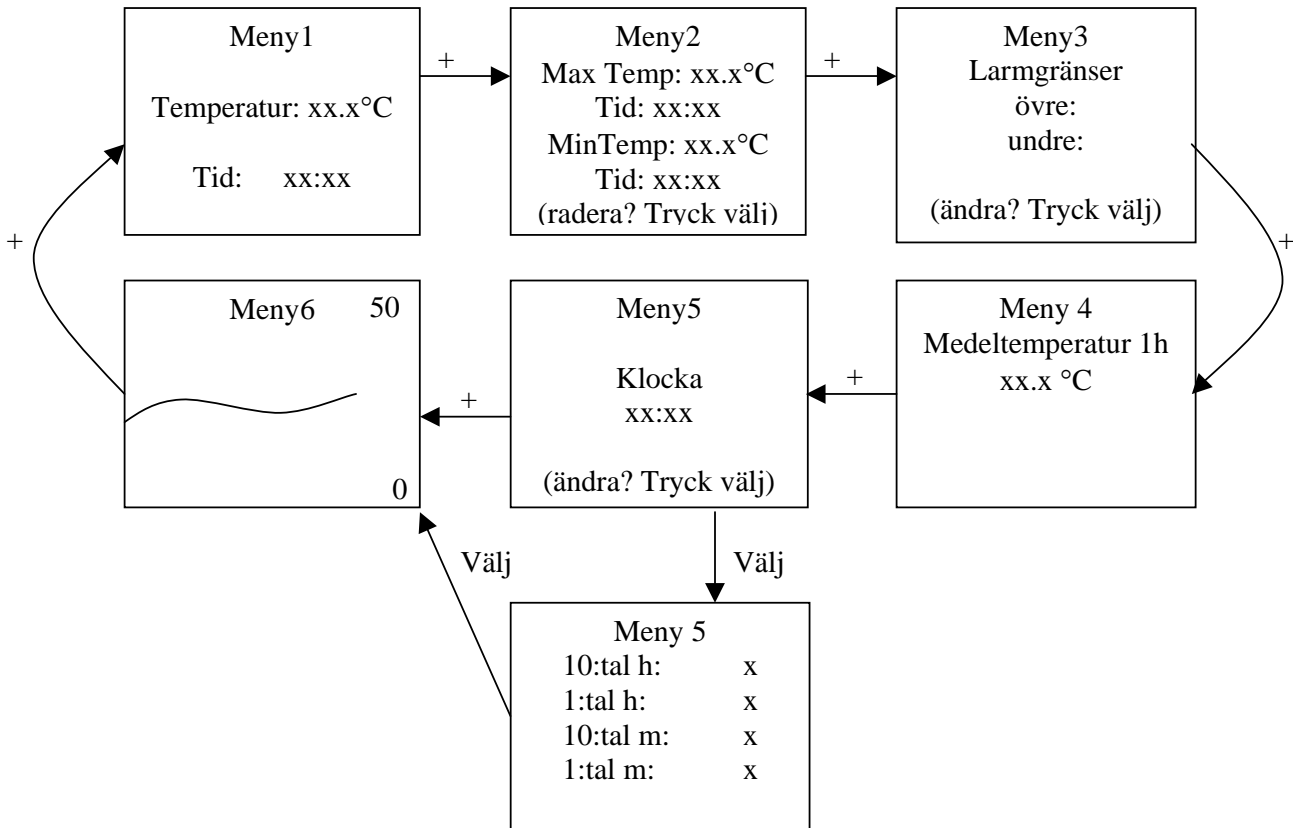
### 2.1 Komponenter

De viktigaste komponenter vi har använt till vårt projekt:

- AVR Mega 16 processor
- AVR JTAG ICE
- 128 X 64 pixels display
- Tryckknapp
- Lysdiod
- Temperaturgivare LM 35
- Förstärkare CA3140E
- Motstånd
- Trim potentiometer

### 3 Mjukvara

Vi har skrivit en mycket kod och bifogar därför den som bilaga. Vårt program har menyer och dom beskrivs av blockschemat nedan.



Figur 2 Blockschemat över programmet

Det första som händer när systemet slås på är att man hamnar i meny 1 där dyker det upp en tid som är satta som start tid. Det finns även en temperatur som automatiskt uppdateras var 10:e sekund. Trycker man på plus så kommer man vidare till meny 2 som blockschemat visar i figur 2. I meny 2 visas maximal registrerad temperatur och vilken tid den uppmättes, samt motsvarande för mintemperatur. Här kan man trycka på mittenknappen som heter välj om man vill nollställa dessa två temperaturer. I meny 3 Ställer man in gränserna för ett larm som kan indikera när någon av gränserna bryts. Meny 4 tar dom 6 senaste lagrade temperaturerna och bildar ett medelvärde och då vi mäter temperaturen var 10 minuter så är det medelvärdet för den senaste timmen. I meny 5 kan man se klockan precis som i meny ett men här kan man även ställa in tiden om man trycker på välj så att man kommer in i en under meny till meny 5. Meny 6 är våran graf där kan man se alla lagrade temperaturer i ett diagram som på ett överskådligt sätt talar om hur temperaturen har varierat dom senaste 24 timmarna. Varje prick motsvarar en mätning och det är ca 12 minuter mellan varje mätning. Minsta temperaturen är när pricken hamnar längst ner på skärmen, då är temperaturen 0 grader, största är längst upp så klart och där är den 50 grader. I normal rumstemperatur kommer punkterna alltså hamna ungefär mitt i fönstret. Minus

knappen göra att man kommer tillbaka till meny 1 i alla menyer utom meny 1 där den nollställer larmet. I meny 1 kan larmet även nollställas med välj.

### **3.1 Koden**

Vår kod blev ganska omfattande, strax över 1500 rader kod och kommer därför inte att ingå i rapporten. En kort beskrivning av dess innehåll följer däremot här. Vi hade en Mainmetod som har kontroll över hela programmet, till hjälp har den en hel del funktioner som kommer att beskrivas i korthet nedan:

#### **char chargen(int in)**

Gör om int typ till char

#### **void init(void)**

initiering av avbrottsrutinen

#### **void SIG\_OUTPUT\_COMPARE0(void)**

själva avbrottet som ser till att allt som ska göras vid ett avbrott blir gjort.

#### **void busy(int chip)**

kontrollerar om displayen är redo att användas.

#### **void comandLCD (uint8\_t com,int chip)**

används för att skicka kommandon till displayen (ex. display ON)

#### **void writeLCD(uint8\_t wri,int chip)**

Lägger ut data åt displayen och utför en skriv cykel.

#### **void write(char ch, int chip)**

Skriver bokstäver till displayen

#### **void clearline(uint8\_t line,int chip)**

Rensar en rad från allt som finns där.

#### **void fillline(uint8\_t line,int chip)**

Fyller en rad.

#### **void clear(int chip)**

Rensar en displayhalva.

### **void fill(int chip)**

Fyller en displayhalva.

### **void ADConv(void)**

Gör en AD-omvandling och lägger resultatet i en global variabel.

### **void printTemp(uint8\_t y, uint8\_t x, uint8\_t sid, int temp)**

För att lättare skriva ut temperaturer på displayen.

## **4 Resultat**

Projektet har varit ganska tidskrävande därför har vi inte konstruerat lite mer överflödiga funktioner som två temperaturgivare och möjligheten att mäta negativa temperaturer. Vi har dock undersökt möjligheten till detta och konstaterat att det hade gått att genomföra i mån av tid. Vår projekt kan dock mäta en inomhus temperatur och göra flera olika beräkningar av denna temperatur.

## **5 Förbättringsförslag**

Vi hade ett par funktioner som vi hade tänkt konstruera men som var lite överflödiga när det blev ont om tid. Dessa är inte alls omöjliga att göra och vi har undersökt vissa av dem noggrant.

En extra temperatur givare kunde ha konstruerats. Det hade inte varit något större problem att AD-omvandlaren ytterligare en signal då vi hade många lediga pinnar på port A där AD-omvandlaren ligger. Mjukvaran för omvandlingen är dessutom känd.

Negativa temperaturer är lite svårare förändring då vi måste ha dels en annan uppkoppling runt temperatur sensors LM 35 och dels göra programmet annorlunda då vi måste jämföra två spänningar med varandra istället för att bara mäta en. Detta kan man göra genom att dels använda den färdiga analoga komparatorn som ingår i AVRn eller mäta två spänningar och med hjälp av mjukvaran jämföra spänningarna.

## **6 Slutsats**

Det har varit en mycket rolig och givande kurs, den enda kursen på LTH som får en att tro att man verkligen kan göra något praktiskt. Kursen ger en bra inblick i vad för problem och möjligheter man har med denna typ av konstruktioner. Vi är nöjda med vår färdiga produkt fast vi inte uppnådde alla funktioner vi hade satt upp i början på kursen.

## 7 Referenser

1. Datablad över våran display [Batron 128 X 64](#) Graphic LCD och dess kontrollchipp [KS0108b/HD61202](#) Graphic LCD-Driver.
2. Datablad över våran processor [ATmega16](#) High-performance AVR 8-bit Microcontroller (Complete).
3. Datablad över våran sensor [LM35](#) Centigrade temperature sensor.