

Sommarstugekoll  
Digital Konstruktion EDA234  
Grupp 2

Names  
Chalmers Tekniska Högskola  
**email addresses**

1 december 2011

# Innehåll

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Introduktion</b>                     | <b>3</b> |
| <b>2</b> | <b>Systemspecifikation</b>              | <b>3</b> |
| <b>3</b> | <b>Systembeskrivning</b>                | <b>4</b> |
| <b>4</b> | <b>Blockschema</b>                      | <b>4</b> |
| <b>5</b> | <b>Block, Funktionalitet</b>            | <b>4</b> |
| 5.1      | Dataväg . . . . .                       | 4        |
| 5.2      | Styrenhet . . . . .                     | 4        |
| 5.3      | Funktionsmoduler . . . . .              | 4        |
| 5.3.1    | DTMF-Modul och MT8880 . . . . .         | 4        |
| 5.3.2    | Ljudmodul och ISD2560P . . . . .        | 4        |
| 5.3.3    | Temperaturmodul och DS18S20 . . . . .   | 4        |
| 5.3.4    | Kontrollfunktioner . . . . .            | 7        |
| 5.3.5    | Gränssnitt . . . . .                    | 7        |
| <b>6</b> | <b>Tillståndsmaskiner</b>               | <b>7</b> |
| 6.0.6    | DS18S20 Seriell Kommunikation . . . . . | 7        |
| 6.0.7    | MT8880 . . . . .                        | 8        |
| 6.0.8    | ISD2560P . . . . .                      | 8        |
| <b>7</b> | <b>Tidsdiagram</b>                      | <b>8</b> |
| <b>8</b> | <b>Felanalys</b>                        | <b>8</b> |
| <b>9</b> | <b>Appendix</b>                         | <b>8</b> |
| 9.1      | Komponentlista . . . . .                | 8        |
| 9.2      | Kretsschema . . . . .                   | 8        |
| 9.3      | PCB-Schema . . . . .                    | 8        |
| 9.4      | Programlistningar . . . . .             | 8        |
| 9.5      | Signallista . . . . .                   | 8        |

## Sammanfattning

Systemet som beskrivs i denna rapport är ett automatiserat temperaturkontrollsystem som är åtkomstbart och styrbart via telefon (DTMF). Systemets huvudfunktionalitet är att informera användaren om de aktuella temperaturerna som läses från två anslutna temperatursensorer, och att ge användaren möjlighet att på avstånd styra (av/på) ett antal externa funktioner, såsom exempelvis värmeelement. Information och kontrolldata utbytes genom en vanlig analog telefonlinje, med hjälp av DTMF: Dual Tone Multiple Frequency.

Ett särskilt exempel på användningsområde är kontroll av uppvärmningen i en sommarstuga. Med hjälp av systemet kan man undvika att temperaturen sjunker under en viss gräns, vilket skulle kunna få konsekvenser såsom att rör fryser. Genom att hålla temperaturen på en lämplig nivå kan man undvika oönskade konsekvenser samtidigt som man minimerar uppvärmningskostnader. 'Sommarstugekoll' löser detta problem genom att erbjuda ett enkelt sätt att hålla koll på och kontrollera inom- och utomhustemperaturer. Om ägaren till en sommarstuga vill göra ett besök ringer han/hon helt enkelt upp sommarstugan och slår på lämpliga element, så att sommarstugan hinner värmas till en behaglig temperatur medans ägaren kör dit.

Systemet självt kräver endast en telefonlinje och en vanlig spänningskälla på 5V. Eftersom det är effektsnålt kan systemet stå på under långa perioder, vilket gör det en lätt och trevlig lösning på alla dina sommarstugeuppvärmningsproblem.

## 1 Introduktion

## 2 Systemspecifikation

|                  |           |
|------------------|-----------|
| Matningsspänning | +5V       |
| Strömförbrukning | 80mA      |
| Någonting        | Någonting |

## **3 Systembeskrivning**

## **4 Blockschema**

## **5 Block, Funktionalitet**

### **5.1 Dataväg**

### **5.2 Styrenhet**

### **5.3 Funktionsmoduler**

#### **5.3.1 DTMF-Modul och MT8880**

#### **5.3.2 Ljudmodul och ISD2560P**

#### **5.3.3 Temperaturmodul och DS18S20**

### **Syfte**

Temperaturmodulen i CPLD'n är ansvarig för att hantera seriekommunikationen med DS18S20-temperatursensorerna, via entrådsbussarna, samt att ge ut de lästa temperaturerna i tecken-belopp-format till ljudmodulen, i syfte att låta den i sin tur spela upp de avlästa temperaturen för systemanvändaren.

### **1-Trådsbuss**

Entrådsbussarna är anslutna till matningsspänning (+5V) genom ett pull-up-motstånd på 4.7kΩ, och ändarna av bussen är anslutna till mastern (CPLD) och sensorn (DS18S20), respektive. Då bussen befinner sig i viloläge dras den hög (svag drivning") av pull-up-motståndet. När information skickas över bussen drar den kommunicerande (sändande) enheten bussen låg genom att driva den med en stark logisk nolla.

### **DS18S20**

Den temperatursensor som används är Maxim DS18S20, som ger temperaturmodulen möjlighet att läsa av temperaturen med nio (9) bitars upplösning. Sensorerna som användas i detta system drivs av en extern spänningskälla på +5V, och kommunicerar seriellt över en entrådsbuss. Seriekommunikationen baserar på att bus-mastern initierar skriv- och läs-luckor. Varje sådan lucka är mellan 60 och 120 us lång. Temperatursensorn initieras genom att mastern driver bussen låg i åtminstone 480 us, vilket följs av att temperatursensorkretsen själv drar bussen låg i 60-240 us, efter en återhämtningslucka på minst 1 us. Efter initieringen väntar sensorkretsen på ett ROM-kommando från mastern, följt av ett Funktions-kommando. Varje sådant kommando är en byte lång, och skickas som LSB-först. Mastern skickar en logisk nolla genom att driva bussen låg under hela skrivluckan.

En etta skickas genom att mastern driver bussen låg under en kort period, 1-15us, och sedan släpper bussen under resten av skrivluckans längd. Mellan varje skriv- eller läslucka måste det finnas en återhämtningsperiod på minst 1us.

Då mastern är klar med att skicka över ROM- och Funktions-kommando, kan (beroende på vilka kommandon som sändes) DS18S20-kretsen svara med aktuell data. På samma sätt som ovan måste mastern här initiera en läslucka genom att driva bussen låg i 1-8us, och sedan släppa den (högimpediv). DS18S20-kretsen svarar på den allokerade läs-luckan genom att antingen hålla bussen låg för att överföra en nolla, eller genom att låta bussen dras hög av pull-up-motståndet för en etta. Under den här tiden (upp till 15 us efter att ha släppt bussen) kan mastern sampla bussen för att läsa av vad DS18S20-kretsen skickat. All data som skickas från temperatursensorn skickas som LSB-först, och i 2-komplementsform.

### **Läscykel, Sammanfattning**

En läscykel består av fyra steg: Initialisering, Kommandon, Läsning och Viloläge.

Initialiseringen består i av att mastern driver bussen låg i 512 us, sedan släpper den. Temperatursensorn svarar med en närvaro-puls genom att driva bussen låg i 106 us, och därigenom bekräftar den sin närvaro på bussen och sin operationella status.

Då mastern detekterat närvaropulsen börjar den överföra ett ROM-kommando (Skip ROM, 0xCC), följt av en kort återhämtningslucka och sedan ett Funktions-kommando (Convert Temperature, 0x44), enligt läs-/skriv-luckemetoden som beskrivits ovan. Skip-ROM-kommandot används i det här systemet eftersom endast en temperatursensor används per entrådsbuss. Därmed finns inget behov av att kunna adressera specifika sensorer på bussen. Convert Temperature-kommandot säger till DS18S20-kretsen att börja konvertera temperaturen, och sedan spara den lästa temperaturen i sitt interna minne för att sedan läsas av mastern. Under konverteringsperioden (upp till 750 ms) kan mastern polla temperatursensorns status genom att kontinuerligt sända förfrågningar genom att dra bussen låg i en kort period (4 us). Temperatursensorn svarar med en nolla sålänge den är upptagen med att konvertera temperatur, och sedan en etta såfort den är klar. Då mastern ser att temperatursensorn är klar, initieras temperatursensorn om och ytterligare ett ROM-Funktions-kommandopar överförs. Dessa är 0xCC (Skip ROM) följt av 0xBE (Read Scratchpad), respektive. Idealt ska temperatursensorkretsen nu vara redo att överföra temperaturdata till mastern från sitt interna 9-bytesminne. Då den ges Read Scratchpad-kommandot börjar DS18S20 sända över innehållet i sitt minne över bussen. Mastern går då in i läsningsläget och börjar sampla datan på bussen genom att driva bussen låg, släppa den och sampla efter 4 us, allt enligt den metod som beskrivs ovan. Mastern samplar de första åtta bitarna data som sänds av DS18S20, sedan

en ytterligare, nionde bit. Den sista biten utgör en tecken-bit, 0 för positiv temperatur och 1 för negativ. Då den nionde biten data lästs ger mastern på nytt en initieringspuls, som säger åt temperatursensorn att sluta sända data. Den nyligen lästa temperaturen placeras av temperaturmodulen i CPLD'n (på CPLD2) på den interna temperaturbussen, och TAv sätts till 1 för att indikera att det finns korrekt data på bussen, enligt vad som efterfrågades av styrenheten.

## Signaler

### In

|      |  |
|------|--|
| DQ0  | Kommunikationstråd till temperatursensor 0                       |
| DQ1  | Kommunikationstråd till temperatursensor 1                       |
| TRd  | Styrsignal från styrenheten: Signalerar att läscykel ska inledas |
| TSel | Styrsignal från styrenheten som väljer temperatursensor (0/1)    |

### Ut

|      |  |
|------|--|
| TAv  | Signal till styrenheten för att signalera att valid data finns på bussen |
| Temp | Den interna temperaturdatabussen till ljudmodulen                        |

## Detaljer

Temperaturmodulen är baserad runt en tillståndsmaskin, understödd av en intern Buffer/MUX-modul samt ett antal interna räknare för att generera de nödvändiga tidsfördröjningspulserna.

## Buffer/MUX

Buffer/MUX-modulen är direkt ansluten till de två DS18S20-Temperatursensorerna som använder entrådsbusskommunikationen. Multiplexern (MUX) används för att välja mellan vilken av de två sensorerna (Sesnor 0 eller Sensor 1) som styrenheten vill kommunicera med, och använder signalen TSel. Buffern är en tri-state-buffer med en Enable-signal, E. När E är satt till 1 så kan mastern använda den av TSel valda entrådsbussen som en utgång för att överföra kommandon. När E är satt till 0 kan mastern läsa data från bussen.

## Räknare

Internt använder temperaturmodulen ett antal olika räknare:

| Namn     | Bitar | Syfte                                    |
|----------|-------|--|
| cntInt   | 9     | Genererar timing-pulser                  |
| ZC       | 4     | Hanterar timing för skriv-luckor         |
| Progress | 2     | Anger aktuellt kommando (0-3)            |
| bitCnt   | 8     | Anger aktuell bit för överföring/läsning |

## Tillståndsmaskin

För en uttömmande beskrivning av den interna tillståndsmaskinen som används av temperaturmodulen, se sektionen om tillståndsmaskiner.

#### 5.3.4 Kontrollfunktioner

#### 5.3.5 Gränssnitt

## 6 Tillståndsmaskiner

### 6.0.6 DS18S20 Seriell Kommunikation

(Refererar till tillståndsdigramet för temperaturmodulen)

**Reset:** Alla signaler och räknare återställs till sina grundvärden.

**Grundvärden:** Utgångspunkten är att alla signaler behåller sina gamla värden om inget annat anges.

#### Initialisering (Tillstånd 0-3):

- 0
- 1
- 2
- 3

Viloläge och återställningspunkt. Temperaturmodulen

#### Kommandoöverföring (Tillstånd 4-8):

- 4
- 5
- 6
- 7

**6.0.7 MT8880**

**6.0.8 ISD2560P**

**7 Tidsdiagram**

**8 Felanalys**

**9 Appendix**

**9.1 Komponentlista**

**9.2 Kretsschema**

**9.3 PCB-Schema**

**9.4 Programlistningar**

**9.5 Signallista**



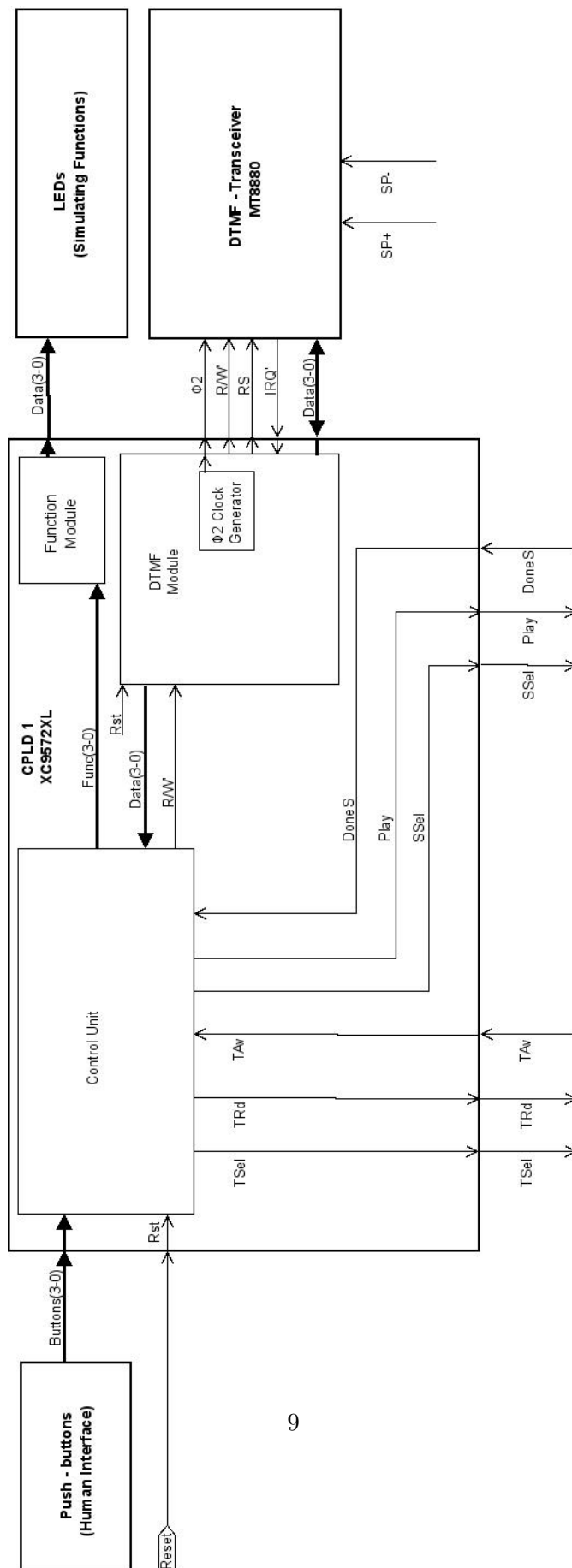
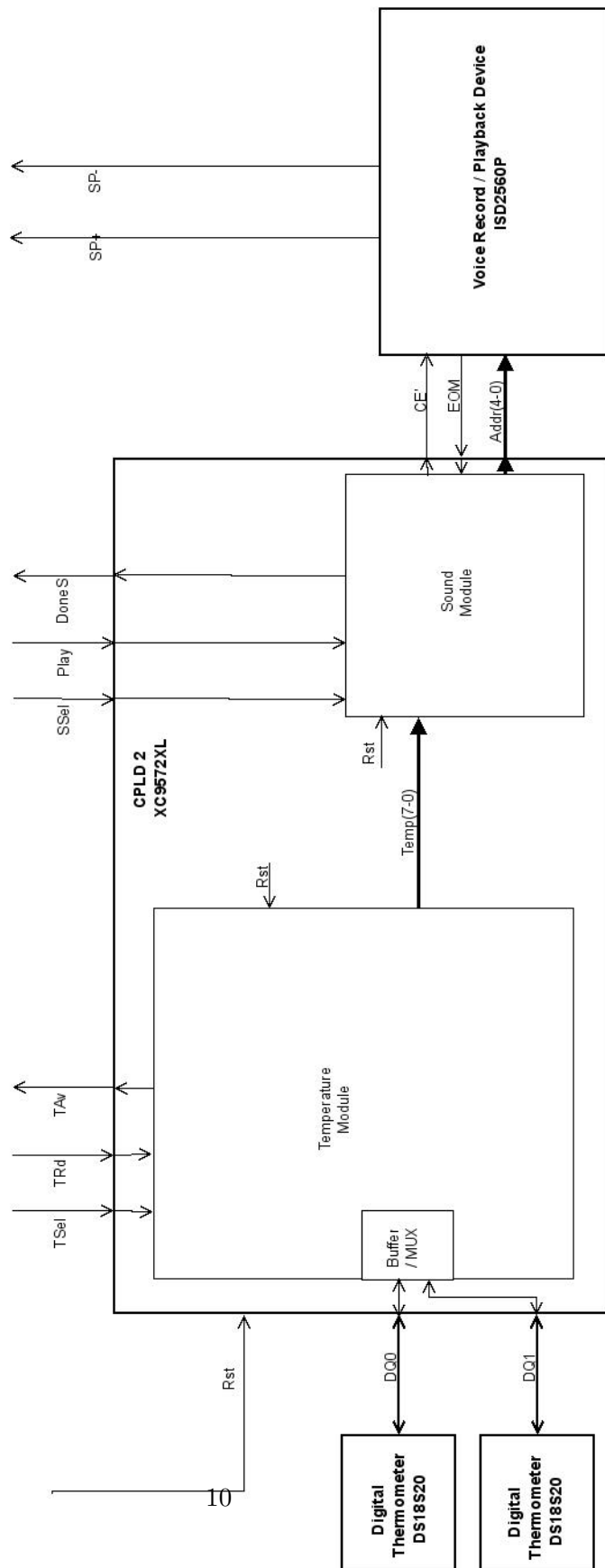


Figure 1: Block Diagram (CPLD1)



Figur 2: Block Diagram (CPLD2)

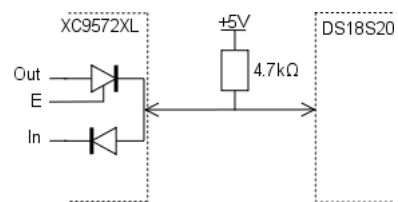


Figure 3: Schematic of 1-Wire bus