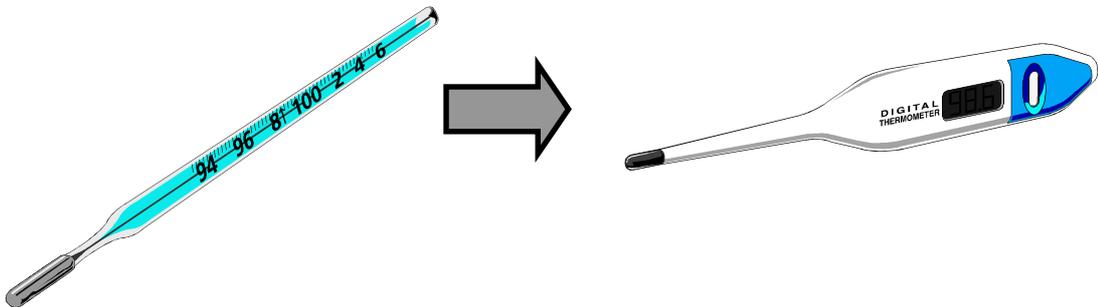


# Projektarbeit Microcontroller

**Thema**

## **A/D Converter**

**mit Hitachi H8/3067**



**Erstellt von**

**Adrian Jetzer André Fleckner Roger Hürlimann**

**Datum**

**November 2001**

# Inhaltsverzeichnis

1. Aufgabenstellung .....	3
2. Aufgabenanalyse (Konzept).....	3
3. Lösungskonzept .....	4
3.1 Hardware .....	4
3.2 Software .....	5
4. Softwarebeschrieb.....	6
4.1 Allgemein .....	6
4.2 Struktogramm .....	7
4.3 Optionen.....	7
5. Bemerkungen/Schwierigkeiten.....	8
6. Schlusswort .....	8

# 1. Aufgabenstellung

Microcontroller: Projektarbeiten

TSU

## Aufgabe 3: A/D-Converter

*Schwerpunkte: HW, Timer, etc.*

Im H8 ist ein A/D –Konverter eingebaut sowie ist auf dem Board ein Temperaturfühler vorhanden. (LM35 IC200)

Erstellen Sie eine SW welche die Temperatur einliest und auf Anfrage auf der seriellen Schnittstelle zur Verfügung stellt.(und im Display kontinuierlich darstellt).

Erweitern Sie die SW so, dass Sie den Wert des LM35 kalibrieren können (wenn nötig). Dies bedingt, dass sie eine Methode finden, bei der sie die Temperatur am IC präzis messen können.

Unterlagen: Datenblatt LM35 / Hardware Manuel H8

## 2. Aufgabenanalyse (Konzept)

Gemäss der Aufgabenstellung ist auf dem Microcontollerboard ein Temperaturfühler (LM35) vorhanden. Der Temperaturwert kann mit dem im H8 eingebauten A/D-Wandler eingelesen werden.

Als erstes galt es das Datenblatt des LM35 zu studieren. Anschliessend stand die Analyse des A/D-Wandlers an. Aus dem Datenblatt des H8 konnten wir die Parameter für die A/D-Wandlung evaluieren.

Ebenfalls überlegten wir wie die Ausgabe auf die RS232 Schnittstelle zu realisieren wäre.

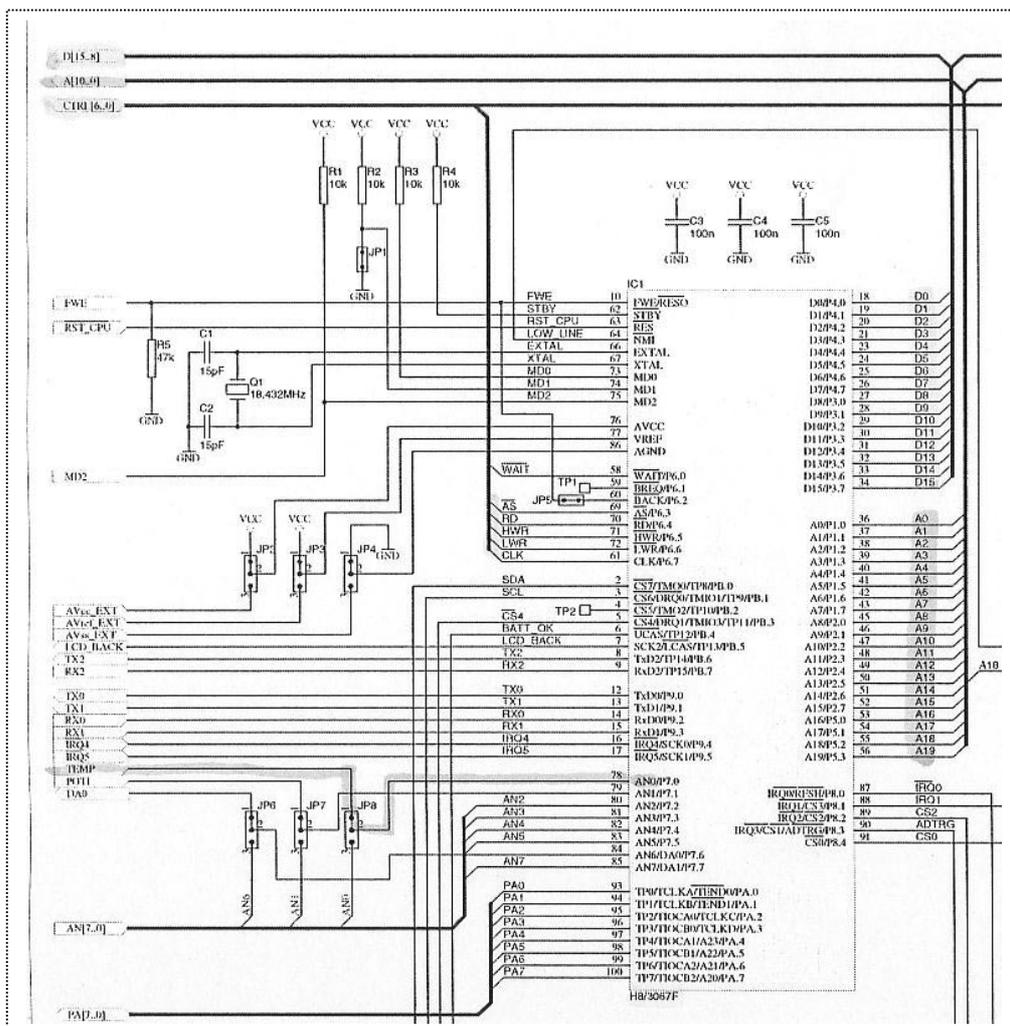
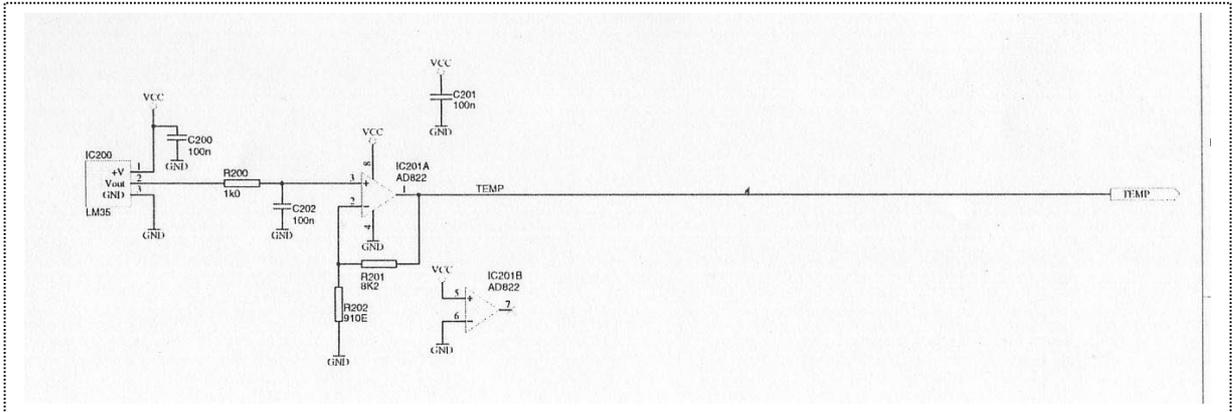
Nach dem Studium der Datenblätter konnten wir erkennen, dass eine Kalibrierung des Systems nur durch eine externe Referenztemperatur möglich ist. Dies könnte mit Hilfe eines Klimaschranks erfolgen. Mit dieser Messung könnte die Abweichung ersichtlich gemacht werden und in der Software mit einem entsprechenden Offset versehen werden.

Als zusätzliche Features könnte die Umrechnung von Grad Celsius in Kelvin und Fahrenheit eingebaut werden.

### 3. Lösungskonzept

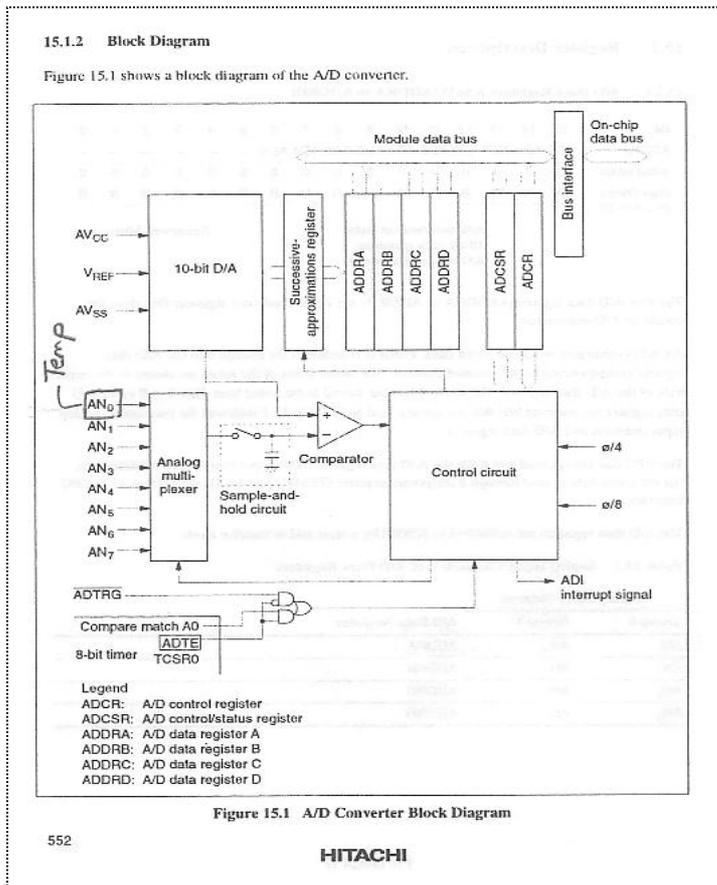
#### 3.1 Hardware

Das Anschlussschema zeigt, dass der LM35 am Eingangport PIN78 (AN0/P7.0) angeschlossen ist. Der Jumper (JP8) muss wie im Schema gezeichnet gesetzt werden (Pin1,2 verbunden).

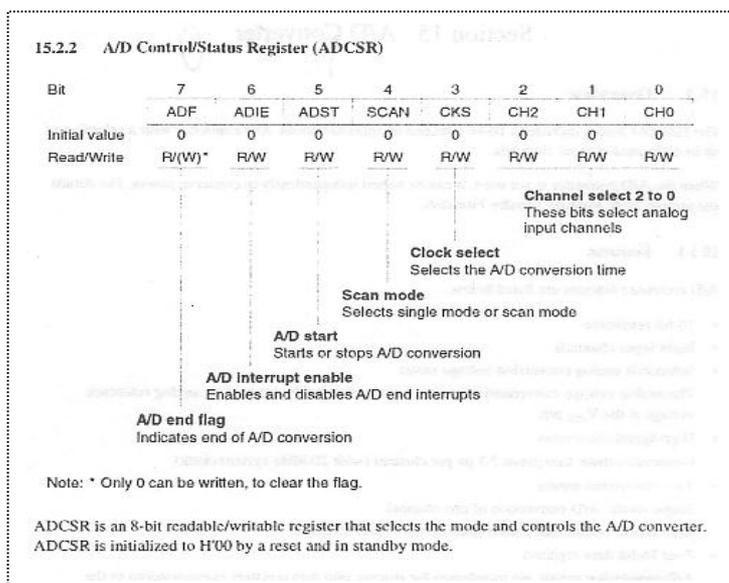


### 3.2 Software

#### Ansteuerung des A/D-Wandlers:



Aus dem Blockschaltbild ist ersichtlich, dass der Wert der Temperaturmessung über den Eingang AN0 nach der A/D-Wandlung in das Register ADDR A geschrieben wird. Damit dies ohne Fehler funktioniert müssen diverse Parameter vorab initialisiert werden.



In unserem Fall ergibt das für das Register ADCSR folgende Parameter:

ADF (Bit 7)	0
ADIE (Bit6)	0
ADST (Bit5)	1
SCAN (Bit4)	0
CKS (Bit3)	0
CH2 (Bit2)	0
CH1 (Bit1)	0
CH0 (Bit0)	0

=> Binär : 00100000      => HEX      : 0x20

ADF (A/D End Flag)	: Setzt Flag wenn A/D-Wandlung beendet
ADIE (A/D Interrupt Enable)	: Auf 0 setzen wenn keine Interrupt Abfrage
ADST (A/D Start)	: Startet die Wandlung
SCAN (SCAN-Mode)	: Selektiert Single- oder Scan-Mode
CKS (Clock-Select)	: Selektiert die Wandlungszeit
CH0-CH2 (Channel-Select)	: Selektiert die Eingangskanäle

## **4. Softwarebeschrieb**

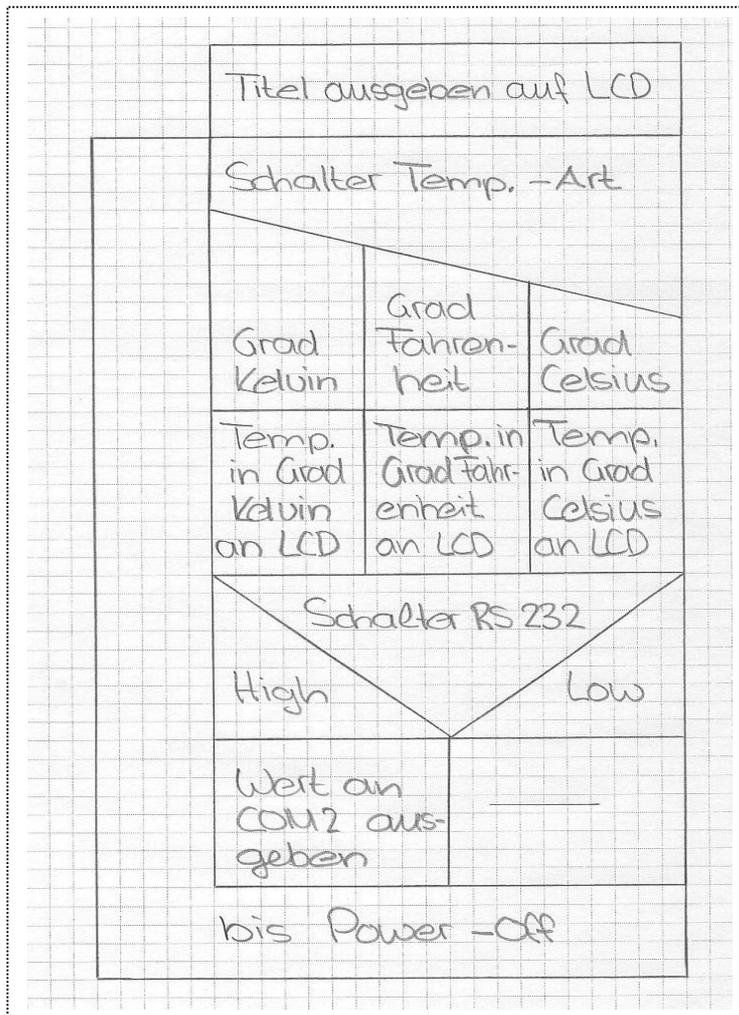
### **4.1 Allgemein**

Auf der ersten Zeile werden im Wechsel unsere Namen angezeigt. Auf der zweiten Zeile kann der Name des Projekts abgelesen werden (Temperaturmessung).

Auf der dritten Zeile wird die aktuelle Temperatur ausgegeben. Mit dem Schalter 1 kann auf Kelvin mit dem Schalter 2 auf Fahrenheit und mit dem Schalter 3 auf Celsius umgeschaltet werden.

Wenn der Schalter 0 auf high geschaltet wird, kann über die RS232 mit dem Hyperterminal in Windows98 die Temperaturen erfasst werden. Es ist denkbar dass die ausgelesenen Werte in Excel oder Matlab weiter bearbeitet werden können.

## 4.2 Struktogramm



Die Beschreibung der einzelnen C-Befehle kann direkt aus dem Source-Code entnommen werden. Der Source-Code ist im Anhang abgelegt.

## 4.3 Optionen

Wie in der Aufgabenanalyse erwähnt, könnte eine Temperaturkalibrierung des Systems eingebaut werden. Dies könnte wie folgt realisiert werden:

- Die Temperaturabweichung mit Hilfe eines Klimaschranks (z.B. exakt 20°C) genau messen
- Ein Schalter auf high legen
- Die Temperaturabweichung über die Tastatur (z.B. +0.3°C) eingeben, wobei 'A' Plus und 'B' Minus bedeutet
- Schalter auf low legen und kontrollieren ob die Temperatur nun genau stimmt

Die Eingabe der Temperaturabweichung könnte auch nur mit Schaltern realisiert werden.

Aus zeitlichen Gründen konnten wir dieses Feature nicht mehr in die Software integrieren.

## **5. Bemerkungen/Schwierigkeiten**

Es ist unbedingt zu beachten, dass das End-Flag, das durch den A/D-Wandler gesetzt wird, zuerst abgewartet wird. Ansonsten wird, wie wir selber bemerkt haben, ein falscher Wert ins Register eingelesen.

Der Temperaturwert wird in das Register ADDRA geschrieben. Dieses 16 Bit Register ist in zwei 8 Bit Register aufgeteilt, da die Register nur durch einen 8 Bit Datenbus verbunden sind.

d.h. die unteren 8 Bit müssen nicht beachtet werden, da die hintersten 6 Bit reservierte Bits sind die immer als 0 ausgelesen werden und die vorderen zwei ohnehin in einem Bereich liegen der viel zu genau ist.

Den verkürzten Wert muss anschliessend noch durch 10 geteilt werden um das Komma an die richtige Stelle zu setzen.

## **6. Schlusswort**

Wir konnten bei diesem Projekt alle profitieren. Obwohl der C-Code nicht sehr umfangreich aussieht, sind doch ein paar Knacknüsse enthalten.

Was wiederum unterschätzt wurde ist die Dokumentation. Viele Gedanken oder Diskussionen wurden nicht auf Papier festgehalten, was die Erstellung der Dokumentation erschwerte.

**Adrian Jetzer**

**André Fleckner**

**Roger Hürlimann**