

## Vereinfachte Temperaturmessung mit mehreren DS18B20 ohne Kenntnis der ROM-Codes in C

Die Temperaturmessung mit den DS18B20 Sensoren ist an sich eine sehr komfortable Angelegenheit. Etwa lästig ist allerdings, dass die ROM-Codes der einzelnen Sensoren bekannt sein müssen, um die gemessenen Temperaturwerte einem Sensor zuordnen zu können.

Spontan einen Sensor auszutauschen oder hinzuzufügen, das ist daher nicht möglich.

Einfacher wird es, wenn man nur einen einzelnen Sensor betreibt. Nicht nur, dass bei einem einzelnen Sensor völlig klar ist, wer die Temperatur gemessen hat.

Auch der Messvorgang selbst ist einfacher, da nach einem "SKIP ROM" nur das Scratchpad ausgelesen werden muss, die Kenntnis des ROM-Codes des Sensors ist nicht erforderlich.

In einer überschaubaren Umgebung (ich benötige nie mehr als 8 Sensoren an einem Controller) kann man die Verwaltungsarbeit vereinfachen, sofern man für jeden benötigten Sensor einen Portpin frei hat.

Denn wenn man immer nur einen DS18B20 (genau genommen natürlich seinen DQ-Pin) an einen Portpin anschließt, dann kann man ihn in der oben beschriebenen, vereinfachten Form auslesen.

Man muss lediglich dafür sorgen, dass die low-level Routinen zur Kommunikation mit dem DS18B20 gezielt auf unterschiedliche Pins zugreifen können.

Im Beispielcode benutze ich an einem ATmegax8 6 Pins des PORTC, um bis zu 6 Sensoren anzusprechen. (An PORTB hat man 8 Pins frei, aber hier ist auch der Programmieradapter angeschlossen).

In der Header-Datei muss zuerst vereinbart werden, über welchen Port und über welche Pins die Sensoren angeschlossen sind.

Das Auslesen der Temperatur läuft dann wie folgt ab:

Zuerst wird ein "CONVERT" verschickt, wobei alle freigegebenen Pins von PORTC dieses Kommando gleichzeitig an alle angeschlossenen Sensoren senden.

Daraufhin führen alle Sensoren ihre Messung aus.

Nach Ablauf der Konversionszeit von ca. 800ms wird nacheinander an jedem einzelnen (der freigegebenen) Pins des PORTC ein "SKIP ROM" und ein "READ SCRATCHPAD" ausgeführt und der Messwert umgerechnet.

Und schon sind alle Temperaturwerte der angeschlossenen Sensoren ermittelt.

Und ich kann genau definieren: "Der Sensor an Pin0 hat den Wert xy geliefert".

Wie aber kann man erkennen, wenn kein Sensor angeschlossen ist oder wenn er nicht antwortet ?

Ist an einem freigeschalteten Pin kein Sensor angeschlossen, dann liegt der Pin - bedingt durch den Pullup - immer auf High, das Programm liest daher immer 0xFF als Datum ein.

Der umgerechnete Temperaturwert beträgt dann  $-0.1^{\circ}$ . Dieser Wert kommt natürlich auch im realen Leben vor, als Indiz für einen nicht antwortenden Sensor taugt er also nicht.

Nun gibt es das Configurations Register, das als Byte 4 aus dem Scratchpad gelesen wird.

Hier ist das höchstwertige Bit immer gelöscht. Diese Kenntnis wird genutzt:

Liefert (Byte 4 & 0x80) ein TRUE, dann ist kein Sensor angeschlossen.

Das Programm gibt als Fehlerhinweis den erkennbar abwegigen Wert -99.9 aus.

Das hier beschriebene Verfahren hat den erforderlichen Programmcode (gegenüber meinen bisherigen Lösungen) deutlich reduzieren (das wesentliche Programmcode benötigt knapp unter 800 Byte) und hat das Verwalten von ROM-Codes überflüssig gemacht.

Allerdings muss man für jeden DQ-Pin jeweils einen Pullup spendieren !

Michael S.