1-wire Temperaturmessung mit DS18B20 und TINY2313 (ROM-Search-free)

Hallo allerseits,

ja - fertige Lösungen zu diesem Problem gibt es bereits, das ist mir bekannt.

Eigentlich wollte ich auch nur eine der Vorlage kopieren und adaptieren.
Aber das wäre vermutlich mühsamer - auf jeden Fall weniger lehrreich gewesen, als auf der Grundlage der umfangreichen Application Notes von Maxim das Programm gleich in der eigenen Denkweise zu schreiben.

Das 1-wire-Protokoll ist ja eigentlich sehr einfach zu verstehen, lediglich das ROM Searchen (oder besser die Implementation dazu) ist schwer verdauliche Kost. Ausserdem treten Netzwerkprobleme zuerst beim ROM-Searchen auf, wenn alle Geräte gleichzeitig an der Datenleitung herumpullen (siehe dazu AN 148, S. 6).

Daher war ausdrückliches Ziel der Übung, einerseits ohne ROM-Search zu arbeiten und andererseits die Grundfunktionalität auf einem möglichst schlanken Controller lauffähig zu bekommen.

Das angehängte Programm kommt mit weniger als 2048 Byte Flash aus, ein TINY2313 reicht damit aus.
Die Aufgabenstellung für den Anfang: Temperaturdaten angeschlossener Sensoren einsammeln, konvertieren und per serieller Schnittstelle an einen Logger weiterreichen.

Der 1-wire-master arbeitet nach folgendem Prinzip:

- Die Steuerung und Datenausgabe erfolgt über die serielle Schnittstelle.

- Jeder Sensor muss einmalig angemeldet werden (dabei wird sein ROM-Code im Eeprom des Controllers gespeichert).
Dazu wird der Sensor exclusiv (!) am Bus angeschlossen und anschließend das Kommmando 0x31 über die serielle Schnittstelle gesendet.
Sind alle Sensoren (nacheinander einzeln) angemeldet, dann werden sie zusammen auf den Bus gesteckt - und die Messungen können beginnen.

- Messungen werden zeitgesteuert durch ein SKIP ROM/CONVERT an alle angeschlossenen Sensoren angestoßen.

- Anschließend werden die Sensoren nacheinander mit ihren Adressen angesprochen, ihr Scratchpad ausgelesen, die CRC geprüft, die gemessenen Temperaturwerte in ein lesbares Format gebracht und über die serielle Schnittstelle ausgegeben.

In der Datenausgabe entspricht die Reihenfolge der Messwerte der Anmeldereihen-folge der Sensoren.

Der 1-wire-master kann über 1-Byte Kommandos via serielle Schnittstelle gesteuert werden:

- 0x10 -> Messungen starten, die Messungen werden in einstellbarem Zeitintervall wiederholt.

- 0x20 -> Messungen beenden.

- 0x30 -> Den ROM-Code eines einzelnen (einzigen !) Sensors ausgeben.

- 0x31 -> Wie 0x31, aber zusätzlich wird sein ROM-Code im Eeprom ablegen.

- 0x40 -> Die ROM-Codes aller Sensoren aus dem Eeprom löschen.

- 0x50 -> Die ROM-Codes aller angelernten Sensoren über RS232 ausgeben.

- 0x60 -> Autostart einschalten (die Messungen beginnen sofort nach dem Einschalten), der Modus wird im Eeprom gespeichert.

- 0x70 -> Autostart abschalten (die Messung muss mit dem Kommando 0x10 gestartet werden), der Modus wird im Eeprom gespeichert.

- 0x8x -> das Messintervall in Sekunden (*10) festlegen.
Das Kommando setzt sich aus dem Bit.7 (128) sowie der dazu addierten Zeit (von 1 - 127) zusammen. Die Zeit wird mit 10 multipliziert und definiert das Messintervall in Sekunden.
Das Messintervall wird im Eeprom abgelegt.

Die im Eeprom gespeicherten Werte werden nach einem Reset wiederhergestellt.

Pro Sensor werden 8 Byte gespeichert, die Größe des Eeproms begrenzt die Anzahl der anschließbaren Sensoren.
In den ersten 8 Byte wird das Messintervall und der Autostart-Modus abgelegt.

Auf einem TINY2313 kann man 15 Sensorcodes speichern.
Sollen mehr benötigt werden, dann greift man zum ATMEGA 48/88 mit 256/512 Byte Eeprom für 31 bzw.63 Sensoren.

Auch sollte es nicht schwer sein, nach einem Umzug auf einen ATMEGA ein LCD zur Temperaturausgabe anzuflanschen.

Für die Eingabe/Ausgabe der Daten habe ich mit HTerm gearbeitet.

Für die Eingabe ist es ratsam, in den Input-controls als 'Type' das Format "HEX" zu wählen.
Alternativ müsste man zu jedem Kommando das entsprechende ASCII-Zeichen suchen - das wäre lästig.

Eine LED an PD.5 habe ich übrigens noch zur Kontrolle spendiert:

Solange Messungen durchgeführt werden (natürlich auch während der Pausen zwischen den eigentlichen Messungen), blinkt die LED.

Bei gestoppter Messung zeigt sie Dauerlicht.

Um Problemen mit dem Eeprom vorzubeugen, wird EEAR immer zurück auf 0 gestellt, eine Speicherstelle, in der keine genutzten Daten stehen.
Weiterhin ist es sinnvoll, die Brown-out Detection in den fuses zu aktivieren.

Die Beschaltung ist einfach mit Worten zu beschreiben:

Bedingt durch die Verwendung der seriellen Schnittstelle ist ein Quarz notwendig, hier vorgesehen mit 3.686400 MHz.

Der 1-wire-Bus wird an PB.0 angeschlossen, ein Pullup mit 4.7k nach VCC.

Für den Betrieb der Sensoren mit einer 2-Draht-Verbindung sind keine Vorkehrungen getroffen, hier muss in der Funktion OW_convert_all() die Stromversorgung der Sensoren während des Messvorgangs realisiert werden.

Die Kontroll-LED wird an PD.5 angeschlossen (über 1.5k gegen VCC geschaltet).

Das war's schon.

Achja, wenn der ATMEGA zum Einsatz kommen soll, dann ist für das Eeprom noch eine Anpassung für Eeprom-Größe > 256 erforderlich, auch bei der seriellen Schnittstelle haben einige Register andere Namen.

Viel Spaß beim Messen,

Michael S.