

Aufgabenstellung

Zur Aufzeichnung von mehreren Temperaturwerten in gleichmäßigen Zeitintervallen benötigte ich einen Logger, der über einen hinreichend langen Zeitraum netzstromunabhängig arbeitet.

In der Bastelkiste hatte ich noch einige PCF8583 (RTC) und AT24C512 (Eeprom) sowie DS18B50 als Temperatursensoren.

So entstand das Konzept des hier vorgestellten low-current Datenloggers auf der Basis von Standard TWI-Komponenten.

Das verwendete TWI-Eeprom mit 512kBit (= 64KB) reicht aus, um in 5 Minutenabständen jeweils 6 Temperaturwerte (zuzüglich 4 Byte Zeiteintrag) über die Dauer von 14 Tagen aufzuzeichnen. Bei Bedarf kann der Speicherplatz durch zusätzliche Eeproms erweitert werden.

Die Ruhestromaufnahme liegt bei ca. 40uA (bei 4.5 Volt) bzw. ca. 20uA (bei 3.3V).

Während der Messung werden für die Dauer von 1 Sekunde ca. 15mA aufgenommen.

Bei einem Messintervall von 5 Minuten entspricht das einer mittleren Stromaufnahme von 15mA / 300, also etwa 50uA.

Ein Batteriesatz mit 1000mAh sollte somit für eine Betriebsdauer von ca. 10.000h = ca. 400 Tage ausreichen.

Funktionsweise

Der Mega88 übernimmt die Kontrolle und den Dialog mit dem System via serieller Schnittstelle.. Im Konfigurationsmenü können Datum, Uhrzeit und Messintervall eingegeben werden, der Mega88 reicht die Daten via TWI an den PCF8583 weiter.

Nach dem Start des Loggens geht der Mega88 in den Powerdown-Modus, der Controller und die TWI-Komponenten haben dieser Phase eine sehr geringe Stromaufnahme.

Der PCF8583 läuft als Clock (zur Fortschreibung von Datum und Uhrzeit) und als Timer, der nach festgelegten Zeitintervallen seinen Pin INT auf LOW zieht und damit den M88 aus dem Powerdown-Modus aufweckt:

Der Mega88 stößt die gewünschten Messungen an, schreibt die Messwerte als Datensatz mit einem Zeitstempel auf das TWI-Eeprom 24C512 - und legt sich wieder zur Ruhe.

Datensatz

Die Datensatzlänge ist abhängig von der Anzahl der zu speichernden Messwerte.

Für meine Anwendung habe ich 16 Byte gewählt, davon werden 4 Byte für den Zeitstempel benötigt. Bleiben also 12 Byte für Nutzdaten.

Der "Platzbedarf" der Daten ist abhängig vom Werteumfang der Messwerte.

Bei meiner Anwendung habe ich die Temperaturwerte aus Bequemlichkeit als int16_t abgelegt.

Da die Temperaturen mit 1 Nachkommastelle gemessen werden, wird der Messwert mit 10 multipliziert.

Grundsätzlich sind beliebige Datensatzlängen verwendbar. Allerdings sind die Pagegrenzen (128 Byte) des verwendeten Eeproms zu beachten: die automatische Incrementierung der Adressen erfolgt nur innerhalb einer Page. Zur Überschreitung einer Pagegrenze muss neu adressiert werden. Darum bieten sich Datensatzlängen von 2^n Byte an (bis maximal 128 Byte).

Die Zeit-Informationen werden in komprimierte Form gespeichert.

Datum und Uhrzeit in 4 Byte im Format:

YYYY.YYYD DDDD.DDDD SSSS.SSSS SSSS.SSSS

Y = das Kalenderjahr, gezählt ab dem Jahr 2000 (maximal bis 2063)

D = der Tag des Jahres, gezählt ab dem 01.01. des Jahres

S = Sekunde des Tages (in 2-Sekundenschritten)

Bedienung

Die Bedienung/Konfiguration erfolgt über die serielle Schnittstelle und ein Terminalprogramm (wie z.B. "HTerm" von Tobias Hammer).

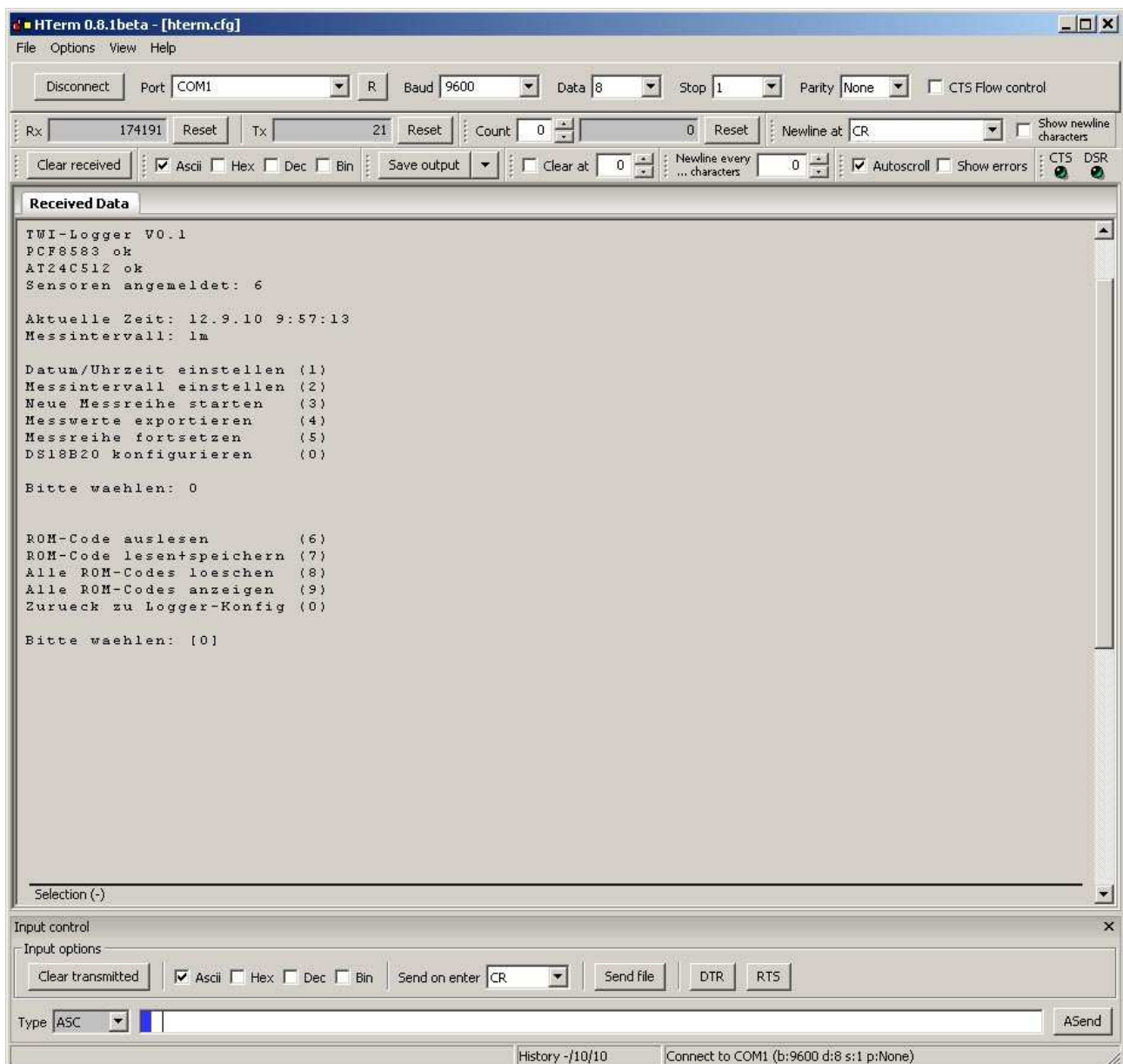
Nach dem Start sollten zuerst Datum und Uhrzeit eingestellt werden, zwingend muss ein Messintervall definiert werden: Die Eingabe erfolgt über die Einheit [Sekunde/Minute] und Dauer (1..99). Danach kann der Logvorgang gestartet werden.

Der Start kann aus dem Menü heraus erfolgen, oder durch Betätigen des Tasters an INT1.

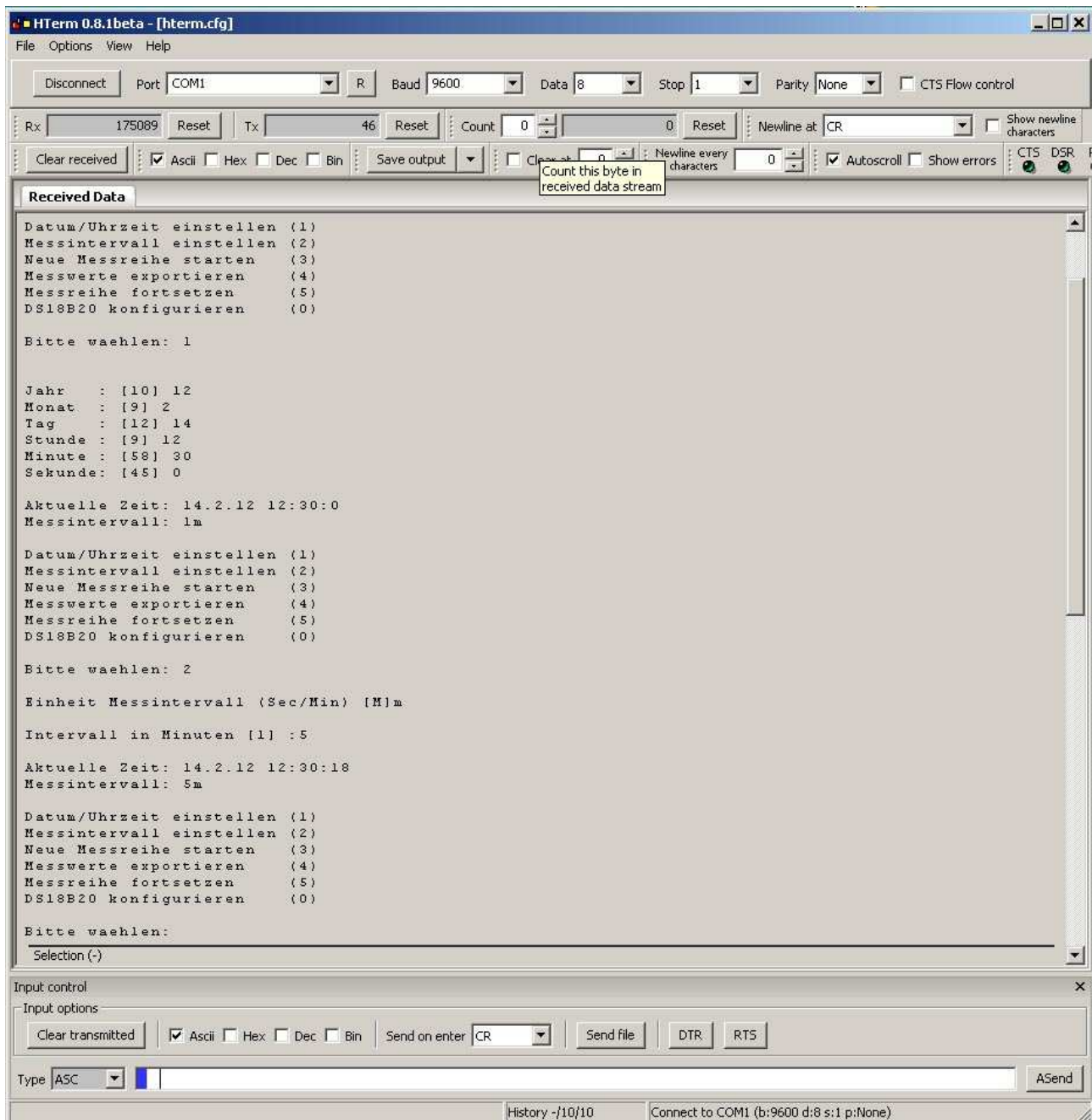
Wird der Pin INT1 während des Loggens über einen Taster auf GND gezogen, dann wird die Messung beendet, das Programm wechselt wieder in das Konfigurationsmenü.

Der Dialog mit dem Programm erfolgt wie in der Abbildung unten dargestellt.

Vorgabewerte werden in eckigen Klammern vorgeschlagen und können mit Return bestätigt werden. Eine Plausibilitätsprüfung für Datum und Uhrzeit wird durchgeführt.



Der Dialog mit dem Programm von oben:
die Meldung beim Programmstart, das Konfigurationsmenü und
das Menü für die Anmeldung der Temperatursensoren



Hier werden im Konfigurationsmenü das Datum, die Uhrzeit und das Messintervall eingestellt.

Für meine konkrete Anwendung (die Aufzeichnung von Temperaturdaten mit DS18B20) gibt es ein weiteres Menü, in dem die Sensoren angemeldet und ihre Adressen im Eeprom des Mega88 gespeichert werden können.

Zur Erläuterung der Temperaturmessung mit den DS18B20 gibt es in Codesammlung einen separaten Beitrag (1).

Bedienung ohne PC

Der Logger kann auch ohne direkte PC-Anbindung rudimentär mit einem Taster an PinD.3 gestartet werden. Aus dem Konfigurationsmenü heraus (LED_KONFIG leuchtet) startet die Taste den Logvorgang (so als wenn ein '5' im Dialog eingegeben würde).

Ein erneuter Tastendruck beendet das Loggen und wechselt wieder in das Konfigurationsmenü.

Damit man sich nicht blind darauf verlassen muss, dass der Logger loggt - und man dann nach 14 Tagen feststellt, dass er es doch nicht getan hat, geben Leds Hinweise auf den aktuellen Programmstatus:

LED_INT

Ist direkt mit dem INT-Pin des PCF8583 verbunden.

Solange das Alarmbit im Controllregister des PCF8583 nicht gesetzt ist, blinkt die Led mit 1Hz.

Das Bit wird erst in dem Moment gesetzt, in dem mit dem Loggen begonnen wird.

Eine blinkende LED_INT zeigt einerseits, dass die Uhr grundsätzlich arbeitet und andererseits, dass kein Logvorgang ausgeführt wird.

Nach dem Start des Loggens geht der INT-Pin auf high (und die LED aus).

Solange, bis ein Timeroverflow stattfindet: Die Led blinkt nun wieder.

Solange, bis die Alarmflags im Controllregister zurückgesetzt werden (das Loggen wieder gestartet wird)..

Im meinem Programm ist die Voreinstellung so gewählt, dass die LED_INT zusätzlich während der Messung für die Dauer des Messvorganges (ca. 1 Sekunde) aufleuchtet (weil das Flag erst am Ende der Messung zurückgenommen wird).

LED_KONFIG

zeigt an, dass sich das Programm im Konfigurationsmenü befindet.

Die Led verlischt, sobald das Konfigurations-Menü verlassen und mit dem Loggen begonnen wird.

Parallel zur LED_KONFIG sollte immer die LED_INT blinken!

Wenn nach dem Start die LED_KONFIG mit ca. 2Hz blinkt, dann war die TWI-Adressierung des PCF8583 nicht erfolgreich und das Programm wird beendet..

LED_LOG

leuchtet beim Start des Loggens für die Dauer von 1 Sekunde auf.

Wenn nach dem Programmstart die LED_LOG mit ca. 2Hz blinkt, dann war die TWI-Adressierung des 24C512 nicht erfolgreich und das Programm wird beendet.

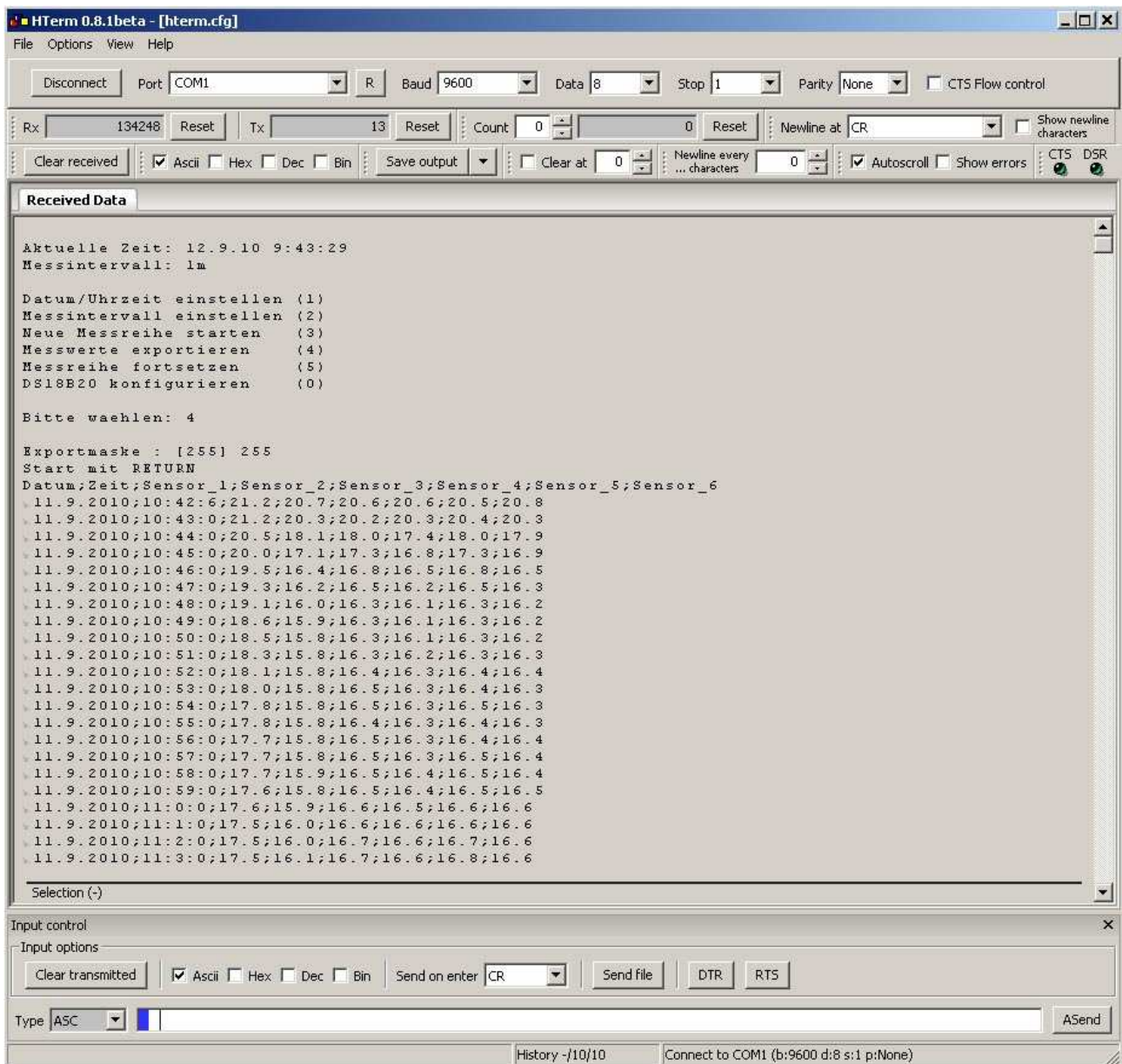
Export von Daten

Zum Auslesen der aufgezeichneten Daten muss eine Konvertierung in ein lesbares Format erfolgen. Um die Daten grafisch mit dem CSV-Viewer von Ingo Wendler darstellen zu können, wird als Ausgabeformat eine CSV-Datei einschließlich der vom CSV-Viewer erwarteten Headerzeile generiert.

Der Import der CSV-Datei auf dem PC erfolgt mit freundlicher der Unterstützung durch das Terminalprogramm "HTerm". Und zwar so:

Wenn im Konfigurations-Menü "Messwerte exportieren" aufgerufen wird, dann wird zuerst nach der "Exportmaske" gefragt. Der Vorgabewert ist 255.

Mit Hilfe der Exportmaske lassen sich die Messdaten einzelner Sensoren aus der Exportdatei ausblenden:



Die Datenbasis mit der Exportmaske '255' exportiert (alle Felder wurden exportiert).

Die Exportmaske

'1' schreibt nur die Messwerte des Sensors_1 in die Datei.

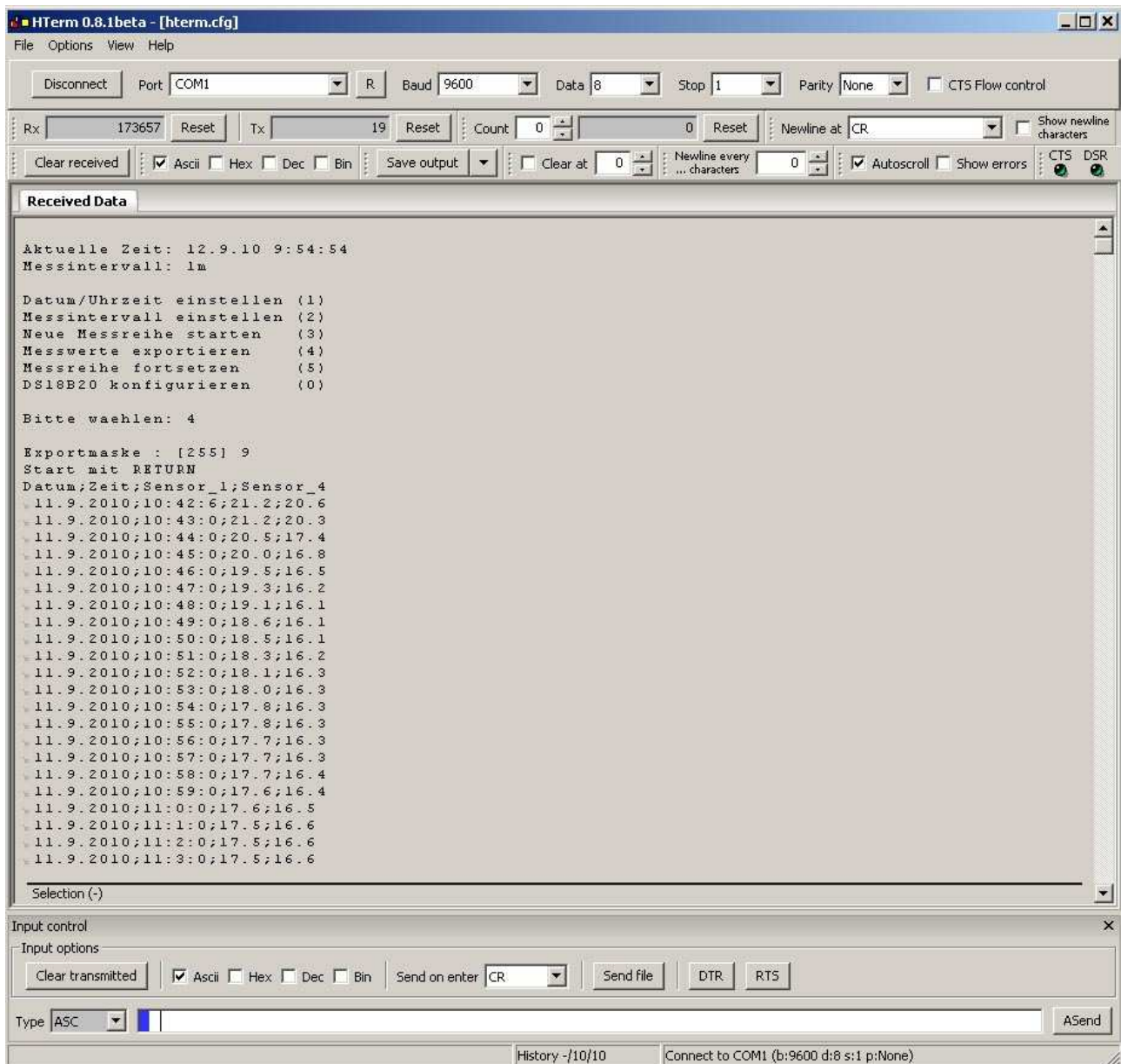
'2' schreibt nur die Messwerte des Sensors_2 in die Datei

'3' schreibt nur die Messwerte von Sensor_1 und Sensor_2 in die Datei

'4' schreibt nur die Messwerte des Sensors_3 in die Datei

'255' schreibt alle Sensordaten ...

Im nachfolgenden Ausdruck wird die selbe Datenbasis wie in obiger Abbildung - allerdings mit der Exportmaske '9' exportiert:



Die Daten mit der Exportmaske '9' exportiert.

Nach der Eingabe der Exportmaske erscheint die Meldung "Start mit RETURN".

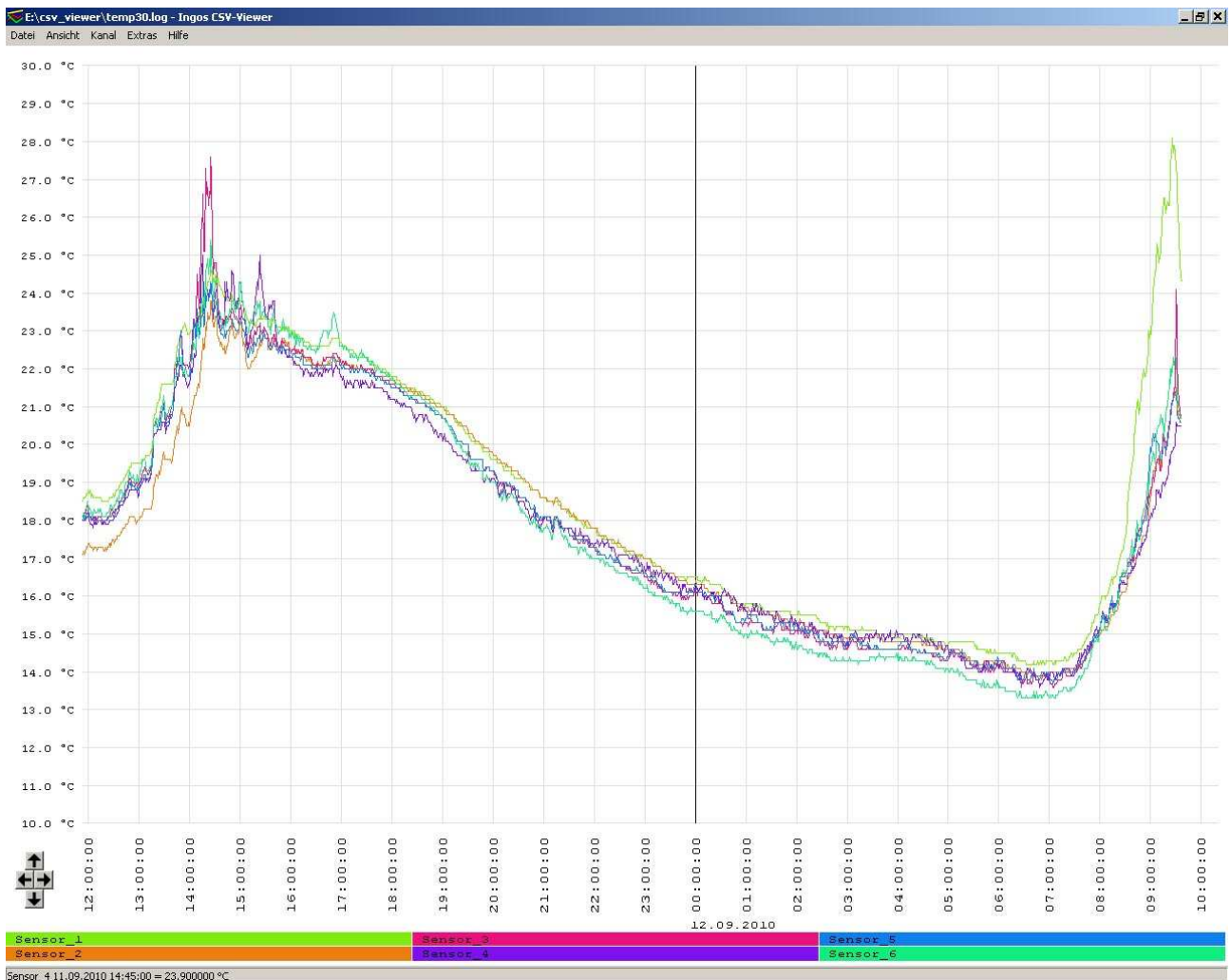
Jetzt lösche ich in "HTerm" den Empfangspuffer mit "Clear received" - und drücke anschließend "Return" am Terminal.

Nun schreibt der Konverter alle Datensätze formatiert auf die serielle Schnittstelle.

Nach dem Abschluss der Übertragung befindet sich der genau Inhalt der CSV-Datei im Puffer von "HTerm".

Von dort wird mit "Save output" der Eingangspuffer in eine Datei auf dem PC geschrieben - und kann von dort direkt vom CSV-Viewer importiert werden.

Mit einem weiteren "Return" am Terminal springt das Programm weiter in das Konfigurations-Menü.



Die exportierten Daten (6 Sensorwerte als Aussentemperaturen) mit dem CSV-Viewer dargestellt.

Tools

Es gibt einige undokumentierte Tools - in erster Linie als Hilfestellung bei der Fehlerbeseitigung. Wer neugierig ist, der findet Hinweise dazu in der `logger_main.c`.

Erweiterbarkeit

Der Logger kann beliebige Daten aufzeichnen, dazu muss lediglich in der Datei `"messung.c"` eine Anpassung an die eigene Problemstellung erfolgen.

Wenn keine DS18B20-Temperatursensoren benutzt werden, dann können die Dateien `"owi_lib.c"`, `"owi_lib.h"`, `"eeprom.c"`, `"eeprom.h"` und `"crc8.c"` gelöscht werden.

Um die Speicherkapazität zu erhöhen, kann man bis zu 2 weitere Chips anschließen und hat damit insgesamt 3 * 64KB zur Verfügung (die Adresse 160 ist bereits fest durch den PCF8583 belegt, daher bleibt nur der Adressraum 162, 164, 166 für die TWI-Eeproms).

Allerdings müsste dazu auch der Adresszähler (`twi_eeprom_adr`) auf 32 Bit erweitert werden. Und beim Lesen/Schreiben - abhängig von der 32-Bit-Adresse - die jeweils zuständige TWI-Adresse gewählt werden.

Diese Änderungen vorzunehmen kann aber nicht wirklich schwierig sein.

Wenn der Logger benutzt werden soll, um auf externe Ereignisse zu reagieren, dann kann man den Pin INT des PCF8583 vom INT0 des Mega88 lösen und eine andere Interruptquelle wählen.

Das "Ereignis" muss nun den INT0 auf GND ziehen und die Messung auslösen.

Allgemeiner Hinweis

Ein Reset des Controllers löscht nicht die relevanten Daten des Loggers:
Datum/Uhrzeit, Messintervall und Eepromadresse des nächsten Datensatzes sind in der Uhr gespeichert, und die Daten im Eeprom sind nicht flüchtig

Wenn die Uhr einen Goldcap oder Kondensator zur Stützung erhält, dann ist auch ein Batteriewechsel möglich, ohne das System neu konfiguriert werden muss.

Wichtig ist, den Adresspin des PCF8583 mit GND (bzw. VCC) zu verbinden.
Tut man das nicht, dann sind lustige Überraschungen vorprogrammiert.

Der Programmcode füllt den Flash-Speicher den Mega88 fast komplett.

Zeitkorrektur

Während der Tests ist mir aufgefallen, dass die Uhrzeit des PCF im Laufe von 24h eine Voreilung von gut 15 Sekunden pro Tag aufwies. Über einen Messzeitraum von 100 Tagen und mehr können so natürlich Zeitabweichungen entstehen, die nicht mehr tolerabel sind.

Daher ist eine softwaremäßige Zeitkorrektur implementiert:

Im Konfigurationsmenü kann man mit 'T' eine Korrekturzeit in Sekunden je 24 Stunden eingeben. Bei einer vorauseilenden Uhr muss die Korrekturzeit negativ sein, taktet die Uhr zu langsam, sind positive Korrekturwerte erforderlich. Der maximale Korrekturwert beträgt +/-127 Sekunden/Tag.

1/24 der Korrekturzeit wird während des Loggens jeweils nach 1 Stunde zur Uhrzeit addiert.

ToDo

Manchmal hatte ich den Eindruck, dass eine kompaktere Datenspeicherung angesichts der knappen Eepromkapazität nützlich wäre.

Man könnte den Zeitstempel in drei Byte verpacken, wenn man sich das Datum des Logbeginns merkt und nur die Tage seit dem Start mitzählt.

Aber dadurch, dass dann kein eindeutiger Zeiteintrag mehr vorhanden ist, lässt sich das Ende eines Datensatzes (nach einem Reset des PCF8583) nicht mehr an der fortschreitenden Log-Zeit rekonstruieren.

Auch die Verwendung von 16Bit Temperaturwerten ist speicherplatzverschwendend.
Dafür aber bequemer im Handling bei wechselnden Anwendungen / Temperaturbereichen.

Mit einem seriellen Eeprom AT45DB321D mit 4 MByte könnte man möglichen Speicherengpässen vorbeugen (2).

Vieles ist machbar - aber eigentlich wollte ich bei diesem Projekt nur bereits vorhandene und bewährte TWI-Komponenten verwenden und ihnen einen würdigen Job verschaffen.

Viel Spaß beim Loggen,

Michael S.

(1) In der "Codesammlung" ist im Beitrag
- "1-wire Temperaturmessung mit DS18B20 für Tiny2313 in C (ROM-Search-free)"
das verwendete Verfahren zur Temperaturmessung beschrieben.

(2) Für eine Erweiterung des Speichers mit einem seriellen Eeprom gibt es ein Programmgerüst in:
- "Datenlogger mit Atmel AT45DB081 + ATMEGA88 in C".

Die Visualisierung mit dem CSV-Viewer von Ingo Wendler habe ich in der Rubrik PC-Hard- und Software im Beitrag

- "Visualisierung von geloggten Daten" gefunden.