Ein Datalogger

Gunther Zielosko

1. Grundlagen

Datalogger sind heute meist kleine unabhängige Geräte, die für sich allein Messungen im Zeittakt ausführen und die Ergebnisse zunächst nacheinander zwischenspeichern. So werden z.B. Temperaturverläufe an unzugänglichen Stellen oder bei Transportvorgängen (Kühltransporte) über lange Zeit erfaßt und im Gerät abgelegt. Später werden sie dann in der Regel mit einem PC ausgelesen und entsprechend dargestellt, bearbeitet oder ausgedruckt. Diese Aufgabenstellung erfordert ein Gerät, das:

- mit Batterie betrieben wird,
- selbst Messungen ausführen kann,
- einen eigenen Speicher hat,
- mit einem Interface zum PC ausgerüstet ist,
- und möglichst eine eingebaute Uhr besitzt.

Sie sehen schon, was herauskommt – das alles könnte der BASIC-Tiger® ohne großen Aufwand erledigen! Trotzdem wollen wir einen ganz anderen Weg gehen; der BASIC-Tiger® soll nicht selbst zum Datalogger werden, sondern ein intelligentes Terminal für einen solchen Baustein. Der Grund dafür ist, daß jede Langzeitanwendung mit Batteriebetrieb für den BASIC-Tiger® mit seinem typischen Betriebsstrom von 50 mA und mehr ein Problem darstellt. In allen anderen Parametern ist der BASIC-Tigers® dagegen eher üppig ausgestattet, wenn wir z.B. nur an die Speichermöglichkeiten denken. Das leidige Stromproblem solcher Anwendungen hat zur Entwicklung spezieller Bausteine geführt, die alle oben genannten Voraussetzungen erfüllen und dabei nur wenige µA benötigen. Allerdings müssen dann Kompromisse bezüglich Speichergröße, Geschwindigkeit und Meßgenauigkeit eingegangen werden.
Ein solcher Baustein ist der DS1616 von Dallas Semiconductor.

1.1. Der Datalogger-IC DS1616

Die wichtigsten Eigenschaften dieses Bausteins sollen hier kurz aufgelistet werden:

- vier 8-Bit Analog-Meßkanäle, von denen einer eine Temperaturmessung mit eingebautem Sensor ausführt,
- eine eingebaute Uhr (RTC) mit Kalender, die sogar Schaltjahre erkennt,
- eine integrierte Schlaf- und Weckautomatik, die RTC-gesteuert Meßvorgänge im Takt von 1 Minute bis 255 Minuten startet,
- einen 2048 Byte großen Speicher,
- ein wählbares Interface (3-Leitung synchrones serielles oder asynchrones serielles – also RS232 kompatibles Interface),
- eine für jedes Chip individuelle Nummer, die zur Identifikation der Meßergebnisse benutzt werden kann,

- verschiedene Meßdaten-Erfassungsvarianten (Datalogger, Histogramm),
- programmierbare Alarmschwellen.

Weitere Angaben können im Internet über

Datenblatt DS1616: http://pdfserv.maxim-ic.com/arpdf/2734.pdf

angefordert werden. Bei Maxim/Dallas kann man auch versuchen, ein kostenloses Muster des (nicht ganz billigen!) IC's zu bekommen (http://dbserv.maxim-ic.com/sl_requests2.cfm).

1.2. Was wir vorhaben

Was halten Sie von einen System mit einem (oder vielen!) Datalogger (n) mit je einem DS1616, die an verschiedenen Orten automatisch die Temperatur und weitere 3 Meßwerte im Minuten- oder Stundentakt erfassen und selbständig mit Uhrzeitangabe speichern. Wenn Sie Zeit haben (oder wenn der Speicher voll ist), gehen Sie mit einem Terminal herum und holen die Daten ab. Anders herum geht es auch, Sie bringen die Datalogger zum Terminal. Das Terminal ist natürlich der BASIC-Tiger® mit seiner seriellen Schnittstelle. Die übertragenen Daten der verschiedenen Datalogger werden hier wiederum gespeichert und können einzeln angezeigt werden. Die dritte Ebene wäre der PC, in den die Meßwerte der Datalogger entweder direkt oder über die Zwischenstufe BASIC-Tiger®-Terminal eingelesen und dann entsprechend komfortabel weiter verarbeitet werden können. Wir werden Hardware- und Softwarevoraussetzungen sowohl für die Auswertung mit dem BASIC-Tiger® als auch mit dem PC schaffen.

2. Die Schaltung

Der DS1616 erledigt die meisten Aufgaben selbst, trotzdem sind einige weitere Komponenten zum Betrieb nötig.
Das beginnt mit der Stromversorgung, der DS1616 braucht zwei verschiedene Betriebsspannungen. Aber keine Angst, es wird nicht so kompliziert, wie man denken könnte. Mit einer geeigneten Batterie, deren Spannung mindestens 2,7 V und maximal 4,5 V betragen sollte, wird der normale Datalogger-Betrieb abgewickelt. Für den (seltenen!) Betrieb der seriellen Schnittstelle wird eine Vcc von 5 V benötigt, die aber dem zweiten an der Kommunikation beteiligten Gerät (BASIC-Tiger® oder PC) „geklaut" wird. Ein Spannungsregler 7805 holt die erforderliche höhere Spannung aus der Schnittstelle, macht daraus 5 V und ein weiterer Baustein (DS275) generiert daraus die Signale für die serielle Kommunikation, ähnlich dem MAX232, der ansonsten solche Aufgaben übernimmt (Plug-and-Play-Lab, in manchen BASIC-Tigern® integriert oder in unserem RS232-Adapter eingesetzt). Der Vorteil des DS275 ist, daß er extrem sparsam mit der Energie aus der seriellen Schnittstelle umgeht und daher ideal für solche „parasitären" Applikationen geeignet ist.

Datenblatt DS275: http://pdfserv.maxim-ic.com/arpdf/DS275.pdf

Weiterhin sind noch ein Quarz mit 32,768 kHz für die Echtzeituhr, einige passive Bauelemente, der 9-polige SUB-D-Stecker und optional ein Taster sowie zwei Leuchtdioden für die Signalisierung verschiedener Betriebszustände notwendig.

Die Gesamtschaltung zeigt Bild 1, die Bilder 2 und 3 zeigen Fotos einer realisierten Variante des Autors.

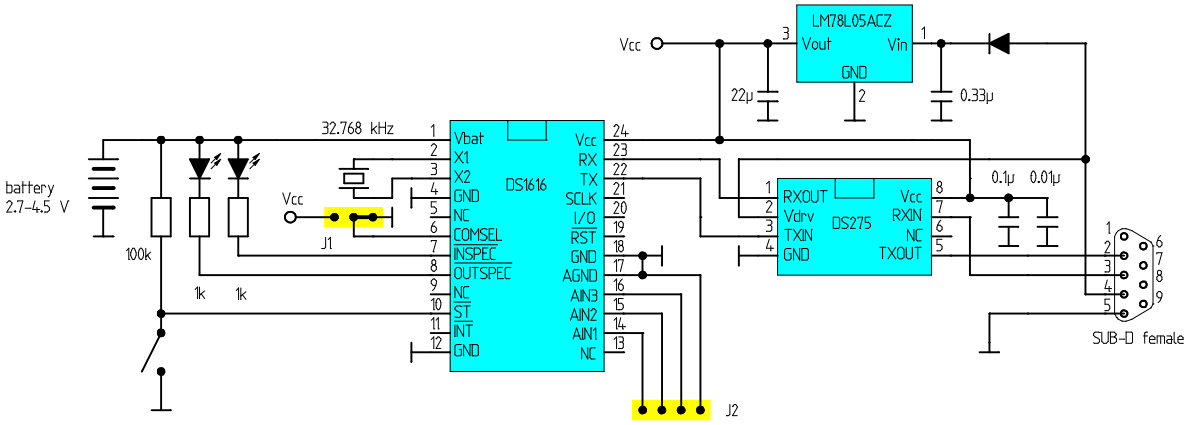

Bild 1 Schaltung des Dataloggers mit DS1616

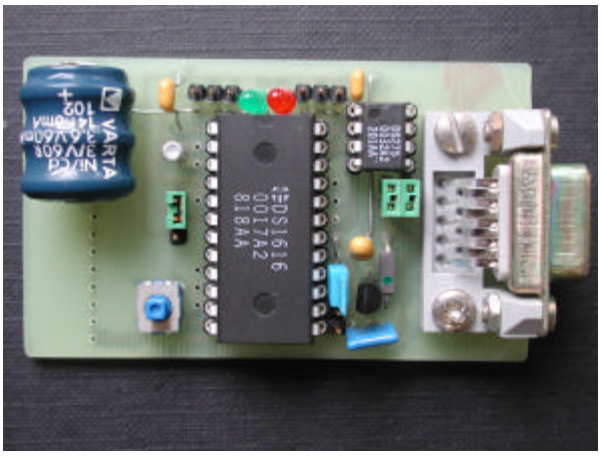

Bild 2 Datalogger mit DS1616

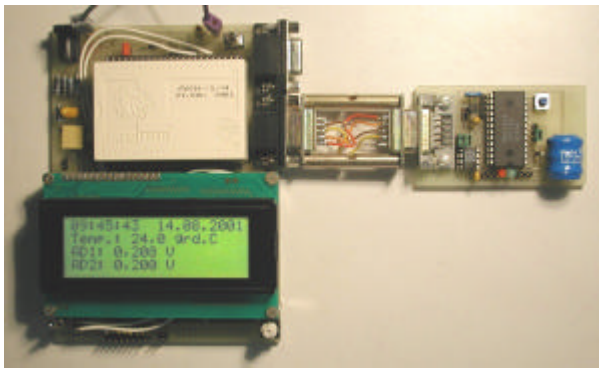

Bild 3 Betrieb mit dem Minilab

3. Erste Inbetriebnahme der Hardware mit dem Demo Programm DS161x.exe

Da unser Datalogger vom Zusammenspiel von Hardware und Software (BASIC-Tiger®, PC) lebt, ist ein erster Test gar nicht so einfach. Er muß zunächst voreingestellt werden, das betrifft z.B. die aktuelle Uhrzeit, das Meßintervall, die Anzahl der benutzten Kanäle, die Art der Datenerfassung und vieles andere mehr. Alle diese Informationen werden über die serielle Schnittstelle an den Baustein gesendet. Der Hersteller des DS1616 bietet im Internet die

kostenlose Demo-Software DS161x.exe mit den dazugehörigen Dateien DS1616.img, mfc42.dll und Msvcrt.dll an, die ebenfalls diesem Applikationsbericht beigefügt wurde. Auf dieser Adresse gibt es sogar den Source-Code in Microsofts Visual C++, nicht unwichtig für perfekte Windows®-Programmierer.

Aktuelle Version: ftp://ftp.dalsemi.com/pub/datalogging/

Mit dem Demoprogramm besitzen wir ein komfortables Tool für erste Experimente mit dem Datalogger DS1616. Kopieren Sie die Exe-Datei mit allen anderen in irgendein Verzeichnis und starten Sie sie dann. Sie werden gefragt, ob Sie mit einem echten oder nur mit einem virtuellen Datalogger kommunizieren wollen. Ist Ihr Gerät fertig, schließen Sie es genauso an eine freie serielle Schnittstelle Ihres PC an wie bisher Ihren BASIC-Tiger®. Wenn Sie nun noch diese Schnittstelle am PC auf 9600 Baud, 8 Datenbits, 1 Startbit, ein Stopbit, keine Parität, kein Protokoll einstellen, können Sie bedenkenlos die Frage von Bild 4 mit einem Mausklick in den jetzt noch weißen Kreis „Read from Device“ beantworten. Wenn es keine Probleme gibt, sollte sich jetzt das Programmfenster öffnen und nach dem Anklicken des linken oberen Buttons erscheint das „Mission Control“- Fenster (Bild 5). Hier tragen Sie fürs erste eine Sample-Rate (Meßintervall in Minuten – hier jede Minute) ein und unten die gewünschten Meßkanäle (1 Temperatur- und drei Spannungskanäle – hier alle). Mit dem zweiten Button „Date/Time“ erreichen Sie die Zeit- und Datumseinstellung. Am besten ist es, die Systemzeit aus Ihrem Computer zu übernehmen. Ist die Zeiteingabe beendet, können Sie wieder in das „Mission Control“ – Fenster zurückkehren und mit dem „Start now“ – Button die Datenerfassung beginnen. Besonders eindrucksvoll ist es, zu Anfang die Messungen mit „DataLog Graph“ (der dritte Button oben) zu beobachten. Denken Sie daran, mit dem „Refresh“ – Button von Zeit zu Zeit neue Daten einzulesen.
Wenn das alles funktioniert, können Sie den Datalogger nun nach Belieben von der Schnittstelle abziehen, z.B. in das Tiefkühlfach legen und am Abend schauen, wie sich die Temperatur im Minutentakt entwickelt hat (Bild 6). Das Ganze geht auch mit den 3 Analogeingängen, deren Spannung im Bereich zwischen 0 V und ca. 2 V liegen sollte. Nun können Sie mit der DS1616 – Software und Ihrem Datalogger spielen, Alarmgrenzen einbauen, Daten speichern und vieles andere mehr. Sie werden eine Menge Anwendungen für ein solches Gerät finden und vielleicht mehr als einen Datalogger im Einsatz haben. Da kommt ein weiterer interessanter Aspekt hinzu – wie unterscheidet man Daten verschiedener Datalogger? Unser DS1616 macht es Ihnen ganz einfach, jedes Exemplar hat eine „eingebrannte“ Seriennummer, die Sie auf dem Fenster im Bild 5 unten erkennen können.

Bild 4 wenn die Hardware funktioniert, kann es losgehen

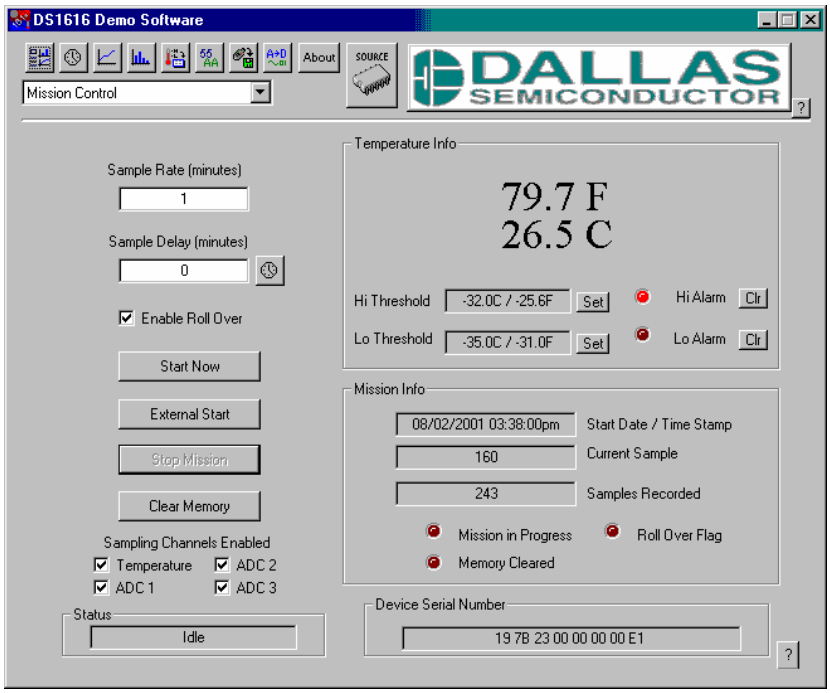

Bild 5 das „Mission Control“ – Fenster

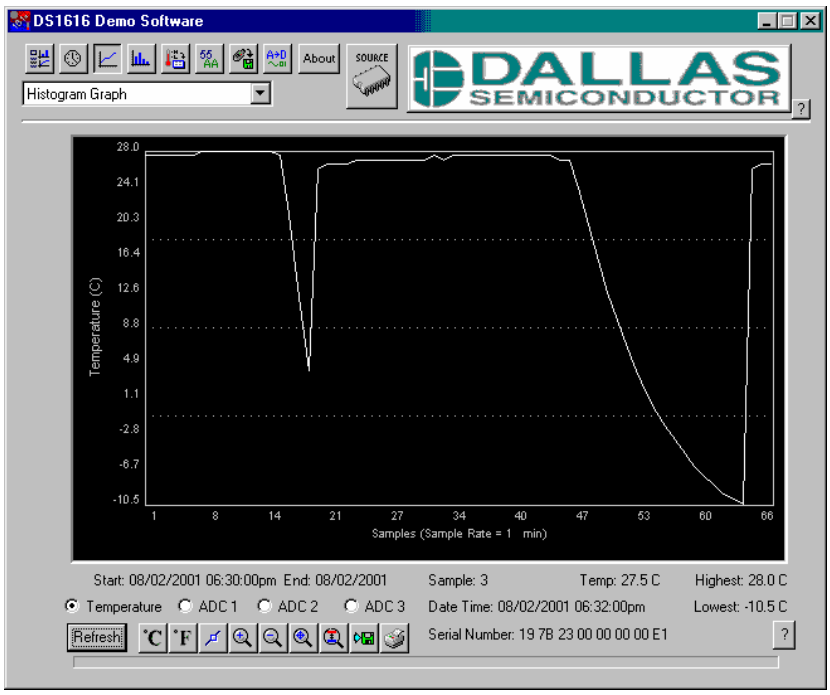

Bild 6 eine aufgezeichnete „Mission“

Die Screenshots wurden der besseren Lesbarkeit beim Druck wegen farblich ein wenig verändert.

Das Programm zusammen mit dem Datalogger ist nicht nur für den Hobby-Elektroniker interessant. Wenn Sie die Eigenschaften des DS1616 z.B. mit Hilfe des Datenblattes voll erfaßt haben, werden Sie staunen über die Möglichkeiten, die in diesem kleinen Baustein stecken. Allerdings können Sie bisher den IC nur mit Hilfe des fertigen Demo-Programms benutzen. Spezielle Betriebsarten lassen sich mit dem Demo-Programm gar nicht realisieren, da muß man selbst programmieren. Leider sind die wenigsten unter uns perfekte Windows®-Programmierer und können sich daher kein eigenes zugeschnittenes Programm schreiben. BASIC-Tiger®-Nutzer haben es da besser, die können mit einem BASIC-Tiger®-Programm die komplette Kommunikation und die Auswertung der Daten genau auf ihre Bedürfnisse hin entwickeln. Und es gibt weitere Vorteile, wie z.B. die Auswertung vor Ort mit einem batteriebetriebenen BASIC-Tiger®. Die Grundlagen dieser Technik werden wir im nächsten Kapitel kennenlernen. Dennoch können Sie das PC-Programm auch dann noch nutzen, um z.B. die Echtzeituhr, irgendwelche Alarmgrenzen oder die Samplerate auf einfache Weise zu programmieren.

4. DS1616 und BASIC-Tiger®

Wenn wir den DS1616 betreiben wollen, kommen wir um einige Kenntnisse seiner Programmierung nicht herum. Ohne auf alle Einzelheiten der internen Register und Datenbereiche einzugehen, hier muß der Interessierte auf das Datenblatt zurückgreifen, werden wir im Folgenden zunächst die Steuer-Befehle kennenlernen. Davon gibt es fünf:

Write Byte	22h
Read Page	33h
Specification Test	44h
Read Data	55h
Clear Memory	A5h

Eigentlich ganz einfach, oder?
Zusätzlich zu diesen Befehlen kommen noch Parameter hinzu, man muß ja angeben, welches Byte beschrieben werden soll usw.

Write Byte (22h)

Der Befehl besteht grundsätzlich aus drei Bytes, dem eigentlichen Befehl, danach eine 8-Bit Adresse (diese kann nur in den unteren 7 Bits variiert werden, das heißt nur 128 Adressen des DS1616 können vom Benutzer eingeschrieben werden) und dem Datenbyte:

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
Befehl	0	0	1	0	0	0	1	0	(22h)
Adresse	0	A6	A5	A4	A3	A2	A1	A0	
Daten	D7	D6	D5	D4	D3	D2	D1	D0	

Dieser Befehl wird gebraucht, um Einstellungen im DS1616 vorzunehmen, Betriebsarten auszuwählen, Vorgänge zu starten bzw. anzuhalten usw.

Read Page (33h)

Auch dieser Befehl besteht aus 3 Bytes, dem Befehl und einer 2-Byte-Adresse:

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
Befehl	0	0	1	1	0	0	1	1	(33h)
Adresse	A15	A14	A13	A12	A11	A10	A9	A8	
Adresse	A7	A6	A5	A4	A3	A2	A1	A0	

Der DS1616 antwortet dann mit der Übertragung einer kompletten Seite (Page) mit 32 Bytes, deren Startadresse die im Befehl Read Page bezeichnete Adresse ist. Welche Informationen wo im Speicherbereich des DS1616 zu finden sind, sagt das Datenblatt. Wir werden später einige wichtige Adressen kennenlernen. Ebenfalls wichtig ist dies: man kann mit dem Auslesen an jeder beliebigen Adresse beginnen, der DS1616 gibt aber nur die Daten der Adressen bis zum Seitenende aus. Das heißt, es werden nur dann komplett 32 Bytes gesendet, wenn die Startadresse auf einen Seitenanfang gesetzt wird (Seitenanfänge sind z.B. 0000h, 0020h, 0040h usw.). Noch ein Hinweis, es wird zusätzlich zu den 32 Bytes ein CRC-Wort (2 Bytes) gesendet, das zur Kontrolle der Übertragung genutzt werden kann.

Specification Test (44h)

Dieser Befehl ist eigentlich unwichtig, er läßt unter bestimmten Bedingungen nur eine der LED’s viermal aufleuchten, so eine Art Funktionskontrolle.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
Befehl	0	1	0	0	0	1	0	0	(44h)

Read Data (55h)

Die augenblicklichen Werte des Temperatursensors bzw. der 3 Analog-Eingänge werden außerhalb der festgelegten Intervalle in spezielle Register geschrieben (current temperature, current ADC-channel)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
Befehl	0	1	0	1	0	1	0	1	(55h)

Clear Memory (A5h)

Die Inhalte praktisch aller Register (Werte, Grenzwerte, Sample Rate usw.) werden gelöscht. Erhalten bleibt lediglich die aktuelle Uhrzeit und einige weitere Daten.

Bit	7	6	5	4	3	2	1	0	
Befehl	1	0	1	0	0	1	0	1	(A5h)

Diese 5 Befehle reichen aus, um alle Einstell- und Lesevorgänge mit dem DS1616 auszuführen.

Bevor wir aber mit dem BASIC-Tiger® loslegen können, brauchen wir noch ein paar winzige Hardwareergänzungen. Der selbstgebaute Datalogger hat eine 9-polige SUB-D-Buchse für den direkten Betrieb am PC, der BASIC-Tiger® auch. Wenn wir beide zusammenbringen wollen, ist ein Adapterstück mit 2 9-poligen SUB-D-Steckern erforderlich, bei denen mindestens die Leitungen 4 und 5 direkt und die Leitungen 2 und 3 über Kreuz verbunden werden müssen (Bild 7):

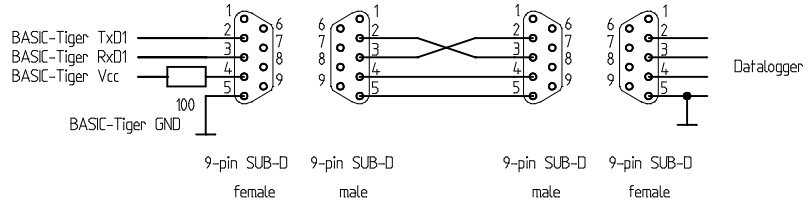

Bild 7 Zwischenstecker für Betrieb DS1616 mit BASIC-Tiger®

Und noch etwas ist zu beachten! Bei einem BASIC-Tiger® mit eingebautem RS232-Adapter muß eine Buchse an SER1 (RxD1, TxD1) angeschlossen werden, deren Pin 4 über einen 100Ω-Widerstand zusätzlich mit der Rohspannung +9V des BASIC-Tiger®-Spannungsreglers 7805 verbunden wird. Analog wird die Buchse SER1 des Plug-and-Play-Lab „präpariert“. Mit dieser Maßnahme wird die Spannungsversorgung des Dataloggers bei der Kommunikation gewährleistet.

Nach diesen Vorbereitungen sollte es klappen.

Im Folgenden werden einige BASIC-Tiger®-Programme vorgestellt, die ein paar wesentliche Funktionen des DS1616 beinhalten. Mit dieser Auswahl sollte es dem BASIC-Tiger®-Programmierer möglich sein, Programme für seine speziellen Anforderungen zu schreiben.

5. BASIC-Tiger® Programme

Wie wir wissen, läuft ohne die Einstellung bestimmter Register im DS1616 gar nichts. Für diese Einstellungen gibt es zwei Möglichkeiten. Entweder wir überlassen das dem PC mit dem Demo-Programm (für die meisten Anwendungen wahrscheinlich die beste Wahl) oder wir tragen Uhrzeit, Sample-Rate, Verzögerung, Alarme usw. per BASIC-Tiger®-Programm ein. Allerdings sollte man ein solches Programm erst dann entwickeln, wenn man den DS1616 genau kennt. Wir gehen hier davon aus, daß der DS1616 vom PC aus initialisiert wurde und bereits eine Mission läuft.

5.1. Abfrage einer kompletten Speicherseite mit PAGE_01.TIG

Dieses Programm liest eine komplette Seite mit der Startadresse 0000h aus dem Speicher des DS1616 und schreibt ihn als Hex-Dump auf das Display des BASIC-Tigers®. Diese Seite enthält die wichtigsten Daten des RTC- und Steuer-Registers, so z.B. die aktuelle Uhrzeit, das Datum sowie weitere Daten, die für uns interessant sind (Bild 8). Natürlich kann der neugierige BASIC-Tiger®-Programmierer auch andere Seiten auf diese Weise untersuchen...

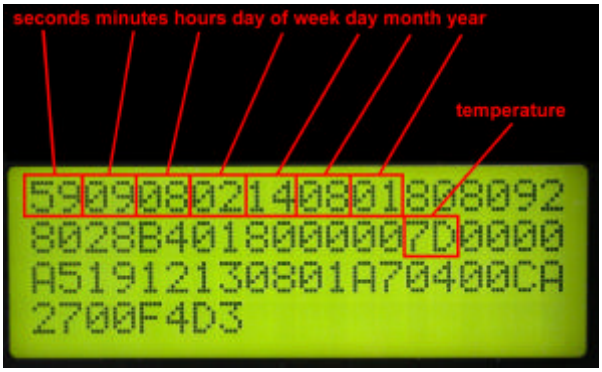

Bild 8 die ersten 32 Bytes aus dem DS1616

5.2. Auslesen von Uhr, Kalender usw. mit TIME_01.TIG

Wer keinen BASIC-Tiger® mit RTC hat, wird die Echtzeituhr des DS1616 schätzen lernen. Völlig unabhängig verrichtet sie, einmal an eine 3,6 V-Batterie angeschlossen und mit dem PC-Programm programmiert, ihre Arbeit. Und das Schönste ist, ihre Zeit muß man nicht mühsam mit vielen BASIC-Zeilen erst aus einem Sekundenzähler herausrechnen. Alle Zeitangaben gibt es in klar verständlichen BCD-Werten, der BASIC-Tiger® muß lediglich die einzelnen Ziffern einsammeln und auf dem Display darstellen (Bild 2 zeigt dieses Programm in Aktion). Bereits diese Eigenschaft des DS1616 bietet tolle Lösungen für die unterschiedlichsten Probleme... Unser Programm liest zusätzlich die aktuelle Temperatur sowie 2 ADC-Kanäle ein.

5.3. Wir schreiben in das User-NV-RAM, das Programm SMS_01.TIG

Vielleicht haben Sie es schon beim Experimentieren mit dem PC-Programm gemerkt, da gibt es eine Möglichkeit, mit der Taste 

zu einem Eingabefeld zu gelangen, über das Sie eine Nachricht im an Ihren DS1616 schicken können. Da der DS1616 als Datalogger möglicherweise sehr lange allein arbeiten muß, kann diese kurze Nachricht (ähnlich einer SMS im Mobiltelefon) z.B. an nachfolgende „Datensammler“ geschickt werden. So weiß dieser, wer vorher kontrolliert hat und wann. Unser BASIC-Tiger®-Programm SMS_01.TIG liest zunächst den Inhalt dieses 32 Byte großen Speichers, schreibt einen neuen kurzen Text in Speicher und liest ihn dann zur Kontrolle gleich wieder aus. Mit dem PC können Sie dann prüfen, ob der Vorgang von Dauer war.

Ein Hinweis: Die BASIC-Tiger® Programme sind im Interesse der leichten Überschaubarkeit sehr simpel aufgebaut. Sie können deshalb problemlos andere Speicherbereiche lesen und beschreiben, so z.B. die interessante individuelle Serien-Nummer, natürlich den bisher nicht berührten eigentlichen Datal-Logger-Bereich usw. Echte Anwendungen mit dem DS1616 werden komplizierter, so sind z.B. die Berechnungen der zu den abgespeicherten Meßdaten gehörenden Zeiten eine echte Herausforderungen für den BASIC-Tiger®-Programmierer (es gibt eine Startzeit und eine Sample-Rate – daraus kann für jedes Sample die zugehörige Zeit ermittelt werden).

6. Anhang: Wichtige Adressen und Datenformate des DS1616

Die Register- und Datenstruktur des DS1616 ist einigermaßen komplex und kann in diesem Rahmen nur auszugsweise dargestellt werden. Der interessierte Anwender sollte sich in jedem Falle das Datenblatt vornehmen und ggf. die folgenden Informationen vertiefen.

6.1. Speicherbereiche DS1616

Adresse	Register Definition	Seite(n)
0000H - 003FH	RTC und Steuer-Register	0 - 1
0040H - 005FH	User NV RAM	2
0060H – 0217H	Reserviert für künftige Erweiterungen	3 – 16 *
0218H - 021FH	Serien-Nummer	16 **
0220H - 027FH	Alarmzeiten und –Zeiträume	17 - 19
0280H - 07FFH	Reserviert für künftige Erweiterungen	20 - 63
0800H - 087FH	Temperatur Histogramm (64 Einträge zu je 2 Byte)	64 - 67
0880H - 08FFH	ADC-Kanal 1 Daten Histogramm (64 Einträge zu je 2 Byte)	68 - 71
0900H - 0FFFH	Reserviert für künftige Erweiterungen	72 - 127
1000H - 17FFH	Datalog-Speicher (64 Seiten)	128 - 191
1800H und höher	Reserviert für künftige Erweiterungen	192 +

* die ersten 8 Bytes
** die letzten 8 Bytes der Serien-Nummer

Tabelle 1: Speicherbereiche DS1616

6.2. Echtzeituhr-(RTC) Register

Bit	7	6	5	4	3	2	1	0	Funktion
Adresse									
00H	0	10er Sekunde			1er Sekunde				RTC-Register
01H	0	10er Minute			1er Minute				
02H	0	12/24	10er Stunde		1er Stunde				
03H	0	0	0	0	0	Wochentag			
04H	0	0	10er Tag		1er Tag				
05H	Y2K	0	0	10er M	1er Monat				
06H	10er Jahr				1er Jahr				

Tabelle 2: RTC-Register

6.3. Steuer- und Status-Register (Auswahl)

Bit	7	6	5	4	3	2	1	0	Funktion
Adresse									
0DH	Anzahl der Minuten zwischen den Messungen (1-255)								Sample-Rate
0EH	EOSC	CLR	0	SE	R0	TLIE	THIE	AIE	Steuer-R. 1
29H	0	CS0	CS1	CS2	CS3	ALIE	AHIE	0	Steuer-R.2
14H	DR	M CLR	MIP	SIP	LOBAT	TLF	THF	ALMF	Status-R. 1
2AH	0	ALF1	AHF1	ALF2	AHF2	ALF3	AHF3	0	Status-R. 2
15H	0	10er Minute			1er Minute				Startzeit der laufenden Mission
16H	0	12/24?	10er Stunde		1er Stunde				
17H	0		10er Tag		1er Tag				
18H	Y2K			10er M	1er Monat				
19H	10er Jahr				1er Jahr				
11H	Aktuelle Temperatur (Bit7...Bit0) , 00H = -40°C, F9 = 85°C								Temperatur
20H	Aktueller Wert ADC1 (Bit7...Bit0), 00H = 0V, FFH = ca. 2V (Ref. Spannung)								ADC1
21H	Aktueller Wert ADC2 (Bit7...Bit0), 00H = 0V, FFH = ca. 2V (Ref. Spannung)								ADC2
22H	Aktueller Wert ADC2 (Bit7...Bit0), 00H = 0V, FFH = ca. 2V (Ref. Spannung)								ADC3

Tabelle 3: Wichtige Daten in den Steuer- und Status-Registern

6.4. Funktion der Steuer- und Status-Bits

Für die richtige Interpretation und die entsprechende Programmierung hier noch die Originalbeschreibungen des Herstellers für die oben erwähnten Steuer- und Statusbits (Auszug aus dem Datenblatt).

CONTROL 1 REGISTER

EOSC - Enable oscillator - This bit controls the state of the oscillator in battery back-up mode only. When set to logic 0, the oscillator is active. When this bit is set to a logic 1, the oscillator is stopped and the DS1616 is placed into a low-power standby mode with a current drain of less than 100 nanoamps at room temperature. When Vcc is applied or when MIP =1, the oscillator is active regardless of the state of this bit.

CLR - Clear Enable - This bit enables the Clear Memory command. When this bit is set to a 1 and the Clear Memory command is subsequently issued, the datalog, histogram, Temperature Alarm, Current Samples, Start Time Stamp, Start Delay, Sample Rate register, and ADC Data Alarm are all cleared to 0. Following the issuing of the Clear Memory command, the CLR bit is also cleared to 0. If the Clear Enable bit is set, but a command other than the Clear Memory command is issued next, the CLR bit is cleared to a 0 and the contents of the datalog, histogram, temperature alarms, Current Samples registers, Start Delay, Sample Rate, and ADC Data alarm register are unchanged.

SE - Start Enable - This bit enables the “start” function of the ST input. When SE is a logic 1, the ST input is enabled as the start pin for datalogging operation. When enabled, datalogging operation begins when the Sample Rate register contains a non-0 value AND theST pin has been held low for at least 0.5 seconds. When SE is a logic 0, writing any non-0 value to the Sample Rate register will start datalogging operation.
Once datalog operation has been initiated, the first data sample occurs after the specified delay written to the Start Delay register has elapsed.

RO - Roll-Over - This bit determines whether the datalog function of the DS1616 rolls over or stops writing data to the datalog memory in the event that the datalog memory is completely filled. If RO is set to a 1, the datalog memory will “roll over” after all 2048 registers in the datalog memory have been used. In other words, after the 2048 th register is written, the following sample will be written to register 0000, overwriting the original data. Likewise, subsequent

samples will increment through the datalog registers, overwriting their data. If RO is cleared to a 0, no further data samples will be written to the datalog memory after all datalog memory registers have been filled. Samples, however, will continue to be taken and the appropriate histogram registers will be incremented with each sample. Likewise, the temperature and ADC Data alarms will also continue to function.

TLIE - Temperature Low Interrupt Enable - When set to a logic 1, this bit permits the Temperature Low Flag (TLF) in the Status 1 register to assert INT . When the TLIE bit is set to logic 0, the TLF bit does not initiate the INT signal.

THIE - Temperature High Interrupt Enable - When set to a logic 1, this bit permits the Temperature High Flag (THF) in the Status 1 register to assert INT . When the THIE bit is set to logic 0, the THF bit does not initiate the INT signal.

AIE - Alarm Interrupt Enable - When set to a logic 1, this bit permits the Alarm Flag (ALMF) in the Status 1 register to assert INT . When the AIE bit is set to logic 0, the ALMF bit does not initiate the INT signal.

CONTROL 2 REGISTER

CSx - Channel Select [0-3] - The value of these bits determines which channels are enabled. A 1 in the CSx bit enables the channel for data collection, recording and reporting. A 0 in the CSx bit disables the channel so data will not be taken, recorded, or reported. This causes a common problem in retrieving data, if the CSx bit is set to 0, the datalog and histogram data will not be downloaded from the DS1616.

ALIE - ADC Data Low Interrupt Enable - When set to a logic 1, this bit permits the ADC Data Low Flag [1-3] (ALFx[1-3]) in the Status 2 register to assert INT . When the ALIE bit is set to logic 0, the ALF bit does not initiate the INT signal.

AHIE - ADC Data High Interrupt Enable - When set to a logic 1, this bit permits the ADC Data High Flag [1-3] (AHFx[1-3]) in the Status 2 register to assert INT . When the AHIE bit is set to logic 0, the AHF bit does not initiate the INT signal.

STATUS 1 REGISTER

DR - Data Ready - This bit indicates the status of the data value in the Current Temperature and/or ADC Data [1-3] registers after the Read Data command has been executed. When this bit is a logic 1, the DS1616 has completed the measurement of all of the selected channels (CSx = 1) and has written valid value(s) to the Current Temperature and/or Current ADC Data [1-3] registers. When this bit is a logic 0, the measurements have not been completed. This bit is cleared to 0 when the Read Data command is sent.

MEM CLR - Memory Cleared - This bit indicates that the datalog memory, histogram memory, Temperature Alarm, ADC Channel 1 Data Alarm, Current Samples, Start Time Stamp, Start Delay, and Sample Rate registers are all cleared to 0. MEM CLR is cleared to 0 when a datalog mission is started (i.e., MIP = 1).

MIP - Mission in Progress - This bit indicates the sampling status of the DS1616. If MIP is a logic 1, the device is currently on a “mission” in which it is operating in the data logging mode. The MIP bit is changed to a logic 1 immediately following 1) the writing of a non-0 value to the Sample Rate register when the SE bit is a 0 or 2) a falling edge on the ST pin if the Sample Rate register contains a non-0 value AND the SE bit is a 1. If MIP is a logic 0, the DS1616 is not currently in datalogging mode. The MIP bit transitions from a logic 1 to a logic 0 whenever datalogging is stopped. Datalogging is stopped when the DS1616 is cleared via the clear bit and clear instruction or when any of the RTC or Control registers (with the exception of the Status registers) are written to during a mission. The MIP bit can also be written to a logic 0 by the end user to stop datalogging. It cannot, however, be written to a logic 1.

SIP - Sample in Progress - This bit indicates that the DS1616 is currently in the process of acquiring a

temperature and/or ADC sample. When the SIP bit is 0, a data conversion is not currently in process and the next conversion will not begin for at least 250 ms. When the SIP bit is a 1, a data conversion is in progress and NO registers or memory locations should be read or written. The SIP bit will be a 1 for a maximum of 750 ms.

LOBAT - Low Battery Flag - This bit reflects the status of the backup power source connected to the V BAT pin. A logic 1 for this bit indicates an exhausted lithium energy source.

TLF - Temperature Low Flag - A logic 1 in the Temperature Low Flag bit indicates that the temperature is/has been less than or equal to the value in the Temperature Low Threshold register. If TLIE is also a logic 1, the INT pin will go low. TLF is cleared by writing this bit to a logic 0. The Clear Memory command has no effect on this bit.

THF - Temperature High Flag - A logic 1 in the Temperature High Flag bit indicates that the temperature is/has been greater than or equal to the value in the Temperature High Threshold register. If THIE is also a logic 1, the INT pin will go low. THF is cleared by writing this bit to a logic 0. The Clear Memory command has no effect on this bit.

ALMF - Alarm Flag - A logic 1 in the Alarm Flag bit indicates that the current time has matched the time of day Alarm registers. If the AIE bit is also a logic 1, the INT pin will go low. ALMF is cleared by writing this bit to a logic 0. The Clear Memory command has no effect on this bit.

STATUS 2 REGISTER

ALFx - ADC Data Low Flag [1-3] - A logic 1 in the ADC Data Low Flag [1-3] bits indicate that the ADC Channel [1-3] Data from the corresponding channel is/has been less than or equal to the value in the corresponding ADC Data Low Threshold [1-3] register. If ALIE is also a logic 1, the INT pin will go low. ALFx is cleared by writing this bit to a logic 0. The Clear Memory command has no affect on this bit.

AHFx - ADC Data High Flag [1-3] - A logic 1 in the ADC Data High Flag [1-3] bits indicate that the corresponding ADC Channel [1-3] Data is/has been greater than or equal to the value in the corresponding ADC Data High Threshold [1-3] register. If AHIE is also a logic 1, the INT pin will go low. AHFx is cleared by writing this bit to a logic 0. The Clear Memory command has no effect on this bit.

Damit sind wir am Ende dieses Applikationsberichtes, aber nicht am Ende mit den Möglichkeiten des DS1616. Dieser interessante Baustein bietet weit mehr als nur Urzeit, Temperatur- und Spannungsmessung. Interessant sind auch bisher nicht oder nur am Rande erwähnte Eigenschaften:

- in das User-NV-RAM können Sie kurze Nachrichten hinterlegen (interessant nicht nur für Wetterforscher in der Antarktis...)
- was ist mit den Speicherbereichen für „künftige Erweiterungen“, kann man die auch nutzen?
- die einmalige und für jeden DS1616 individuelle „eingebrannte“ Serien-Nummer könnte als Zugangskontrolle oder ähnliches genutzt werden.
- welche Sensoren können für die drei AD-Eingänge im Bereich 0...2 V benutzt werden?
- Den Interrupt-Ausgang könnte man vielleicht in Zusammenhang mit den vielfältigen Alarmeinstellmöglichkeiten nutzen (1x pro Sekunde, 1x pro Minute, 1x pro Stunde, 1x pro Tag oder 1x pro Woche gibt es einen Alarm). So könnte z.B. der „schlafende“ BASIC-Tiger® beim Erreichen einer Alarmzeit des DS1616 „aufgeweckt“ werden, alle Daten übernehmen, eine neue Mission starten und dann wieder „schlafen“ gehen.

- Genau dasselbe gilt für die Alarme beim Über- oder Unterschreiten festgelegter Werte. Auch damit könnten Systeme praktisch stromlos lange Zeit arbeiten. Erst wenn Alarme auftreten, wird der BASIC-Tiger® aktiviert und löst irgendwelche Vorgänge aus.

Viel Spaß beim Knobeln...