

---

## **BASIC-Tiger® und GPS**

Jens-Erich Lange, Gunther Zielosko

### **1. GPS – der totale Durchblick**

GPS (Global Positioning System) ist wohl die wichtigste Entwicklung der Menschheit in Sachen Orientierung auf der Erde seit der Erfindung von Kompass und Sextant. Mit seiner Hilfe ist es in den meisten Fällen sogar nautischen Laien möglich, ihre genaue Position zu Wasser, auf dem Lande oder in der Luft zu bestimmen. In den kommenden Abschnitten werden wir kennen lernen, was GPS ist, wie ein typischer GPS-Empfänger funktioniert, was für Daten er anzeigt und ausgibt und wie eine Zusammenarbeit mit dem BASIC-Tiger® aussehen könnte.

Natürlich gibt es auch eine Bauanleitung und ein Beispielprogramm. Der BASIC-Tiger® soll Daten eines fertigen GPS-Empfängers empfangen und in der Art eines Fahrtenbuches eine Reise „archivieren“. Bei Bedarf kann er bei PC-Anwendungen, die einen GPS-Empfänger benötigen, diesen simulieren – auch wenn kein GPS-Empfang möglich ist (z.B. in einem Gebäude). Unser einfaches Fahrtenbuch erlaubt damit die spätere Rekonstruktion einer Fahrt.

#### **1.1. Grundlagen der Satelliten-Navigation**

GPS ist ein komplexes System mit vielen Satelliten und Bodenstationen, das von der Regierung der USA zunächst für militärische Zwecke aufgebaut und später mit der Einschränkung durch „künstlicher“ Ungenauigkeiten (Selective Availability, SA) zivilen Anwendungen geöffnet wurde. Daraus haben sich zahlreiche kommerzielle Anwendungen ergeben, die heute z.B. aus dem Verkehr nicht mehr wegzudenken sind (Navigationssysteme für Schiffe, Flugzeuge, Autos usw.). Am 2. Mai 2000 wurde die Selective Availability durch Präsident Clinton außer Kraft gesetzt, was die Genauigkeit des Systems auf bis zu 5 m (!) Auflösung verbesserte.

Das Prinzip der Satelliten-Navigation ist folgendes:

Eine Reihe von speziellen Satelliten kreist im Weltraum um die Erde. Ein Teil von ihnen ist immer gerade so positioniert, dass ihre Signale an einem gegebenen Standort von einem GPS-Empfänger aufgenommen werden können. Diese Signale enthalten Informationen über den individuellen Satelliten, zur genauen Zeit und zur genauen Position des Satelliten. Aus diesen Angaben wird über komplizierte Berechnungen der genaue Standort des GPS-Empfängers ermittelt.

Prinzipiell erfolgt die Standortvermessung über die Signallaufzeiten von den einzelnen Satelliten, die ihre Position (eigentlich eher ein „Fahrplan“, denn alles geht sehr dynamisch!) gleich mitsenden. Aus der Messung der Signallaufzeiten wird über die Lichtgeschwindigkeit die Entfernung zum jeweiligen Satelliten ermittelt. Bei nur einem Satelliten kann man sich nun vorstellen, dass dies eine Kugel mit dem Radius der gemessenen Entfernung ergibt. Auf der Oberfläche dieser imaginären Kugel muss unser Standort liegen. Werden zwei Satelliten vermessen, erhalten wir zwei Kugeln, unser Standort kann jetzt nur noch auf solchen Punkten sein, in denen sich die beiden Kugeln schneiden (das ergibt einen Kreis). Bei drei Satelliten

bleiben nur noch zwei mögliche Schnittpunkte der nun drei Kugeln übrig, auf denen wir uns befinden könnten. Erst mit 4 Satelliten ergibt sich ein einziger Schnittpunkt, die Feststellung des genauen Standortes einschließlich der Höhe über dem Meeresspiegel ist perfekt.

Das Geniale an GPS ist, dass über Bodenstationen alle Daten der Satelliten einschließlich ihrer Bahnabweichungen immer wieder vermessen werden. Diese Korrekturen werden an die Satelliten gesendet und diese schicken dann wieder Daten mit ihrer „aktualisierten“ Position zurück. Ein GPS-Empfänger enthält neben der komplizierten Empfangstechnik auch ein sehr komplexes Rechenwerk, mit dem die Daten in brauchbare Koordinaten sowie Zeit- und Höhenangaben umgerechnet werden. Wie schwierig allein das Problem des Datenempfanges aus dem Weltraum ist, lässt sich beispielsweise aus folgenden Angaben ableiten:

Insgesamt 24 (mit Reserve-Satelliten sogar 27) umkreisen die Erde mit ungefähr 11200 km/h in etwa 20200 km Höhe. Ihre Sendeleistung beträgt nur 20 – 50 W bei einer Sendefrequenz von 1575,42 MHz. Bei dieser Frequenz herrschen etwa optische Verhältnisse, d.h. jedes Gebäude, jedes Fahrzeug- oder Blätterdach schirmt das Signal weitgehend ab. Elektrische Störungen durch allerlei Funkdienste (Handys usw.) behindern den Empfang zusätzlich. Trotzdem empfangen moderne GPS-Empfänger die Signale von bis zu 12 Satelliten gleichzeitig (!). Zur genauen Standortbestimmung sind mindestens 3 – 4 Satelliten erforderlich.

Aus dem eben Gesagten ergibt sich, dass auch das Wundermittel GPS seine Grenzen hat und man nicht überall und zu jeder Zeit sicher sein kann, dass die Positionsangaben stimmen. Die wichtigsten Fehlerquellen sind:

- Die von den USA zur Zeit abgeschaltete Selective Availability kann, insbesondere in Krisenzeiten, jederzeit wieder in Kraft gesetzt werden. Es ist sogar möglich, dass GPS zeitweise ganz abgeschaltet wird.
- Abschirmung einzelner Satelliten durch Fahrzeugdach, Bäume usw.
- Veränderungen der Signallaufzeit durch verschiedene Bedingungen in der Ionosphäre
- Reflexionen der Signale an Gebäuden oder Bergen und dadurch verlängerte Laufzeiten
- Abweichungen der Uhrzeit (Empfänger – Satelliten)
- Abweichungen der Satelliten-Laufbahnen (treten trotz Korrektur von „unten“ in gewissen Zeitabständen auf)
- Geringe Anzahl von empfangenen Satelliten
- Ungünstige Satelliten – Konstellation (viele direkt über mir, wenige am Horizont)

## **1.2. GPS-Empfänger**

Die Anzahl der mittlerweile verfügbaren Geräte ist riesengroß. Versuchen wir einmal eine Systematik, mit deren Hilfe wir dann zu unserer eigentlichen Applikation kommen wollen:

### **Komplette Einbau-Navigationssysteme**

Solche Geräte kennen Sie von Autos der „gehobenen“ Klasse. Meist fest im Armaturenbrett eingebaut, zeigen sie eine Straßenkarte und die Position ihres Fahrzeuges. Oft gibt es gleichzeitig eine Sprachausgabe, die sie zu ihrem eingespeicherten Ziel bringt. An solchen

---

Geräten gibt es nicht mehr viel zu „basteln“, für ihren Anwendungszweck sind sie meist perfekt, aber auch sehr teuer. Die zur Navigation erforderlichen Kartendaten bekommen sie von einer CD, die im Bedarfsfall gewechselt werden kann. Ähnliche Systeme gibt es auch für Schiffe und Kleinflugzeuge.

### **Navigations-Autoradios**

Ein ähnliches System, nur dass die Kartenfunktionen etwas eingeschränkt wurden. Die Kosten sind schon etwas erträglicher. Auch hier kann man als Bastler nicht mehr viel tun.

### **Handheld-Navigationssysteme**

Eine an die erste Gruppe angelehnte Technik, die allerdings auch außerhalb des Fahrzeugs benutzt werden kann. Die Karten werden hier meist auf kleine Speicherkarten geladen. Das Display ist üblicherweise ein kleiner TFT-Bildschirm. Sehr komfortabel, aber auch recht teuer.

### **GPS-Handys mit Kartenfunktionen**

Kleine, handliche GPS-Empfänger im Handy-Look mit ähnlichen Gebrauchseigenschaften wie die vorherige Gruppe. Die Kartenfunktionen sind vorhanden, haben aber meist eine stark reduzierte Detaildarstellung (z.B. oft kein Farbdisplay, geringe Pixelzahlen).

### **GPS-Handys ohne Kartenfunktionen**

Derzeit die billigste Variante. Diese Geräte beinhalten natürlich einen GPS-Empfänger nebst der dazugehörigen Rechentechnik. Eine Karte kann nicht eingespeichert und demzufolge auch nicht angezeigt werden. Wie manch kompliziertere Geräte auch, haben sie eine Daten-Schnittstelle, mit der die GPS-Daten an externe Geräte (PC, Handheld-Computer, aber auch BASIC-Tiger®!) weitergegeben werden können. Ein einfaches Display, mit dem der Gerätestatus, die Satelliten-Empfangsverhältnisse, Daten wie Position, Geschwindigkeit, Höhe, gefahrene oder gewanderte Strecke angezeigt werden können, ist vorhanden. Man kann Wegpunkte speichern, Routen vorgeben oder Tracks aufzeichnen (d.h. eine Spur z.B. der Wanderung abspeichern). Diese Daten können später am PC wieder abgerufen und z.B. in eine Karte eingetragen werden.

Ein solches einfaches Gerät ist z.B. der eTrex von Garmin, den die Verfasser dieses Applikationsberichtes benutzt haben und hier genauer vorstellen werden.

Alle bisher vorgestellten Geräteklassen sind bereits ohne weitere Peripherie ganz brauchbar und können ganz allein zum Navigieren benutzt werden.

### **GPS-Mäuse**

Es gibt auch Geräte, die nur eine Antenne, einem Empfänger, ein Rechensystem und eine Schnittstelle haben. Allein kann man mit ihnen nichts anfangen, meist haben sie noch nicht

---

einmal ein einfaches Display. Erst in Zusammenarbeit mit einem PC oder eben dem BASIC-Tiger<sup>®</sup> werden sie echt nützlich.

## **GPS-Module**

Einbau-Geräte, die in eigene Konstruktionen integriert werden können. Meist handelt es sich um Miniatur-Module, die eine externe GPS-Antenne benötigen und über einen Steckadapter angeschlossen an andere Komponenten (PC, Microcontroller usw.) werden. Auch ein solches Gerät haben wir getestet und die Hardware dafür ausgelegt. Die Preise solcher GPS-Mäuse oder -Module liegen in der Größenordnung von GPS-Handys ohne Kartenfunktion, häufig sind sie sogar teuer als diese – ein Grund, sich ein deutlich universelleres GPS-Handy zu kaufen, mit dem man auch ohne Peripherie etwas anfangen kann.

## **GPS-Empfänger als PC-Cards**

Diese Sonderform gibt es als Zubehör für Laptops oder Handheld-PC's. Zusammen mit diesen Geräten sind sie äußerst interessant, wegen der komplizierten Schnittstelle aber wenig geeignet für eigene Bastelanwendungen.

### **1.3. Der eTrex von Garmin**

Garmin ist einer der großen Hersteller von GPS-Empfängern aller Art. Schauen Sie ruhig einmal in

<http://www.garmin.com>

vorbei. Eine gute Übersicht über aktuelle Geräte gibt es auch unter

<http://www.mobile-navigation.de/shop/enter.html>

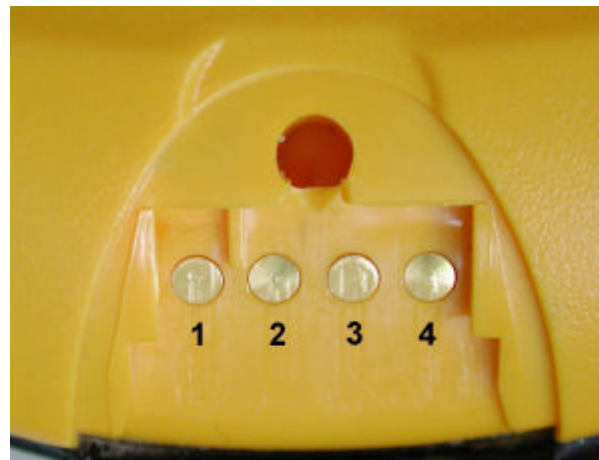
Die eTrex-Familie hat eine Reihe von Mitgliedern, die angefangenen vom Einsteigermodell eTrex über den eTrex Summit (zusätzlicher barometrischer Höhenmesser und elektronischer Kompass), den eTrex Venture (mit Städte-Datenbank) und den eTrex Legend (mit Kartendarstellung). Für uns reicht das einfachste Modell im typischen gelben Gehäuse (Bild 1), denn bereits dieses bringt die typische Garmin-Schnittstelle mit (Bilder 2 und 3).

Im Rahmen dieses Applikationsberichtes können natürlich nicht alle Details dieses kleinen High-Tech-Gerätes dargestellt werden. Wer möchte, kann sich auf verschiedenen Internetseiten oder im Fachhandel informieren.

Wichtig ist für uns später die Hard- und Softwareseite der Datenschnittstelle. Die vom eTrex ist vom RS232-Typ (d.h. Pegel von -3...-12 V und +3...+12 V).



*Bild 1 Garmin eTrex*



*Bild 2 Garmin Connector:  
1 = Power (+), 2 = Data In  
3 = Data Out, 4 = Ground (-)*



*Bild 3 Garmin Connector, Kabelseite*

Der beste Weg, Daten vom eTrex zum PC bzw. unserem Fahrtenschreiber und umgekehrt zu schicken, ist ein optional erhältliches Datenkabel. Man erspart sich damit viel Fummelei mit dem Spezialstecker von Garmin. Beides (lötbaren Stecker oder fertiges Datenkabel eTrex/PC bekommt man bei:

<http://www.haid-services.de/pi1013036774.htm>  
<http://www.haid-services.de/pd-1149209411.htm>

Für diese Verbindungsmethode wurde unser Gerät ausgelegt, es hat einen 9-poligen SUB-D-Stecker zum direkten Anschluss dieses Datenkabels. Mit Hilfe des speziellen Garmin-Steckers können Sie natürlich auch eine eigene Verbindung herstellen, die z.B. die Stromversorgung des eTrex im Fahrzeug mit übernimmt.



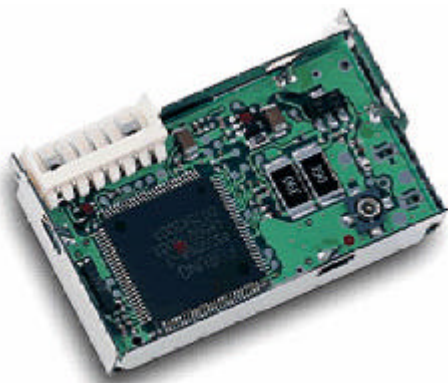
Softwareseitig kann man verschiedene GPS-Protokolle einstellen, wovon wir das NMEA-Protokoll benutzen werden. Dieses Protokoll wird von fast allen GPS-Empfängern beherrscht und fast alle Peripheriegeräte bzw. Software verstehen dieses GPS-Protokoll.

#### **1.4. Der GN79N-NF (5V, BNC)**

Dieses Einbaumodul der Firma Furuno benötigt zum Betrieb eine externe Antenne. Alle Verbindungen (Power, Daten, Pufferbatterie) erfolgen über einen Miniatur-Steckverbinder. Das Modul und Informationen dazu bekommt man bei:

<http://www.hy-line.de/communication/>

In unserem Projekt gibt es einen sogenannten GPS-Connector, der erforderlichen Anschlüsse in der richtigen Reihenfolge enthält.



*Bild 4 Das GPS-Modul Furuno GN79N  
(Foto Furuno)*



*Bild 5 Die dazugehörige GPS-Antenne  
AU-12-5B (Foto HY-LINE)*

## **2. Unser Projekt – Ein automatisches Reisetagebuch**

Ein moderner GPS-Empfänger wie der eTrex kann bereits von sich aus Wegpunkte und Tracks speichern, wenn auch mit Einschränkungen. Wozu dann noch der BASIC-Tiger®? Die Antwort ist einfach, uns reizt der Eigenbau, der freie Zugriff auf die Programmierung und der vergleichsweise enorme verfügbare Speicher des BASIC-Tigers®. Erstmalig können wir selbst entscheiden, welche Daten unseres GPS-Gerätes angezeigt oder gespeichert werden, in welcher Form diese ausgegeben werden und welche Aufgabe Sie insgesamt mit Ihrem neuen Gerät erledigen wollen. Haben wir erst einmal das vorgestellte Projekt realisiert und verstanden, stehen gänzlich anderen Anwendungen von GPS und BASIC-Tiger® nichts mehr im Wege. Dabei soll nicht verschwiegen werden, dass wegen des relativ hohen

Stromverbrauchs des letzteren eigentlich nur Anwendungen im Fahrzeug sinnvoll erscheinen. Aber gerade hier macht es Sinn, eine Art Fahrtenschreiber zu haben.

## 2.1. Die Schaltung

Der Aufwand hält sich in Grenzen. Wir nehmen die kleinste und billigste Bauform des Tigers, einen ECONO-Tiger®, einen MAX232, einen Spannungsregler 7806 (richtig – 6 V!), einen speziellen GoldCap-Kondensator FE0H105Z von Tokin und ein paar weitere Kleinteile. Bild 6 zeigt die Schaltung des einfachen Aufbaues, der sogar auf einer Lochraster-Universal-Leiterplatte realisiert werden kann. Wir haben diesem Applikationsbericht aber auch das Layout einer einfachen einseitigen Leiterplatte beigelegt, das man sich bei Bedarf kopieren und auf fototechnischem Weg in eine fertige Leiterplatte umsetzen kann (Bilder 7 und 8).

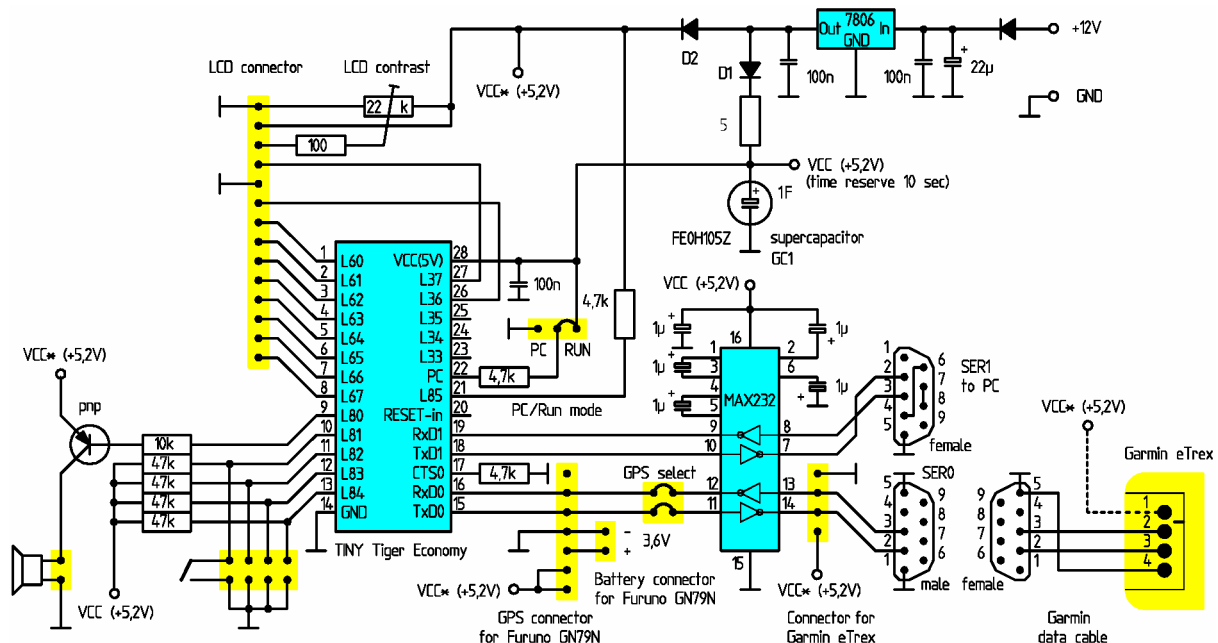


Bild 6 Schaltung der GPS-Fahrtenschreiber-Applikation

Zur Funktion:

Die Schaltung benötigt +12 V und Masse aus der KFZ-Stromversorgung, der Pluspol muss nach dem Zündschalter angeschlossen werden. Der Regler 7806 ist eine 1 A –Ausführung und liefert bei eingeschalteter Zündung am Ausgang 6 V. Über die Diode D1 wird die Schaltung mit der Spannung VCC versorgt, man könnte auch den eTrex mit dieser Spannung arbeiten lassen (siehe Anschlussbelegung eTrex Bilder 2 und 3). Da dieser aber mit 2 Mignon-Zellen recht lange läuft, geht es auch mit interner Versorgung z.B. mit NiMH-Akkus recht gut. Die Diode D1 reduziert diese 6 V auf typisch 5,25 V, gerade richtig für den ECONO-Tiger® und seine Komponenten. Interessant ist der GoldCap-Kondensator GC1 mit 1 F (1 Farad!). Er soll nach Abschalten der Zündung sicherstellen, dass der ECONO-Tiger®, noch ein Weilchen werkeln kann (Daten in den Flash speichern). Bei der Auswahl dieses Kondensators muss man auf einen niedrigen Innenwiderstand von ca. 1 W achten. Die Diode D2 liefert VCC\*,

die den MAX232, ggf. den GPS-Empfänger sowie den aktiven Beeper und die Eingabejumper mit Spannung versorgt. Alle diese Komponenten werden sofort außer Betrieb gesetzt, wenn die Zündung ausgeschaltet wird, um „Reservestrom“ zu sparen. Weiterhin liefert VCC\* einen logischen High-Pegel an L85 des ECONO-Tiger®, der dadurch sofort „merkt“, wenn die Zündung ausgeschaltet wird.

Rechts in der Schaltung befindet sich ein MAX232, der für die richtigen RS232-Pegel an beiden seriellen Schnittstellen sorgt. Die obere SER1 dient wie immer zur (eventuellen) Programmierung des ECONO-Tigers® „on board“ und zum Abfragen von Daten aus dem Fahrtenschreiber. SER0 (unten) ist die Schnittstelle zum eTrex (RS232) oder einem Gerät mit TTL-Pegel (GPS-Maus o.ä.) Die beiden Jumper GPS-Select werden bei Verwendung des eTrex so gesteckt, dass die beiden Leitungen RxD0 und TxD0 über den MAX232 an den 9-poligen SUB-D-Stecker geführt werden. Dann können Sie einfach das Standard-Datenkabel des eTrex zum Anschluss benutzen. Wenn Ihr GPS-Empfänger mit TTL-Pegel arbeitet, lassen Sie die Jumper weg und schließen letzteren an die entsprechenden Stifte des GPS-Connectors an.

Der Port 6 des ECONO-Tigers® sowie einige Steuerleitungen werden für ein optionales Display verwendet. Hier kann der Anwender mit dem Beispielprogramm „GPS-REC.TIG“ einige aktuelle Daten auf seinem 4-zeiligen LC-Display anschauen (Typ und Anschluss wie im Plug-and-Play-System).

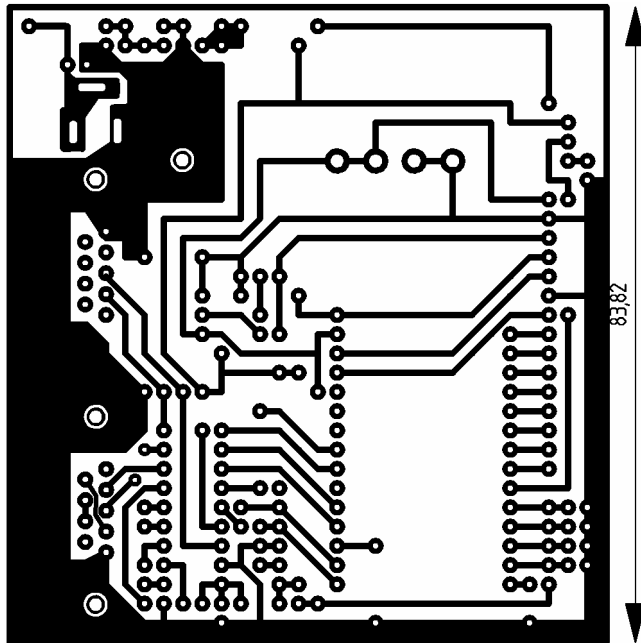
Links unten sehen Sie noch ein paar Bedienelemente, wie eine Taste, einen speziellen Beeper sowie drei Jumper, über die etwas eingestellt werden kann. Zunächst brauchen Sie dort nichts zu stecken.

Die Beschaffung des Super-Capacitors FE0H105Z der Firma Tokin macht etwas Schwierigkeiten. Distributoren sind z.B.:

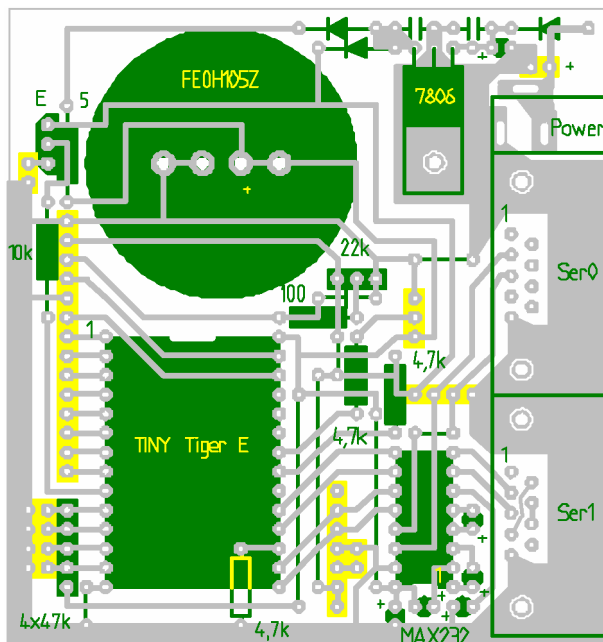
Rutronik <http://www.rutronik.de/>  
Gleichmann <http://www.msc-ge.com/>



## 2.2. Das Platinenlayout



*Bild 7 Layout der einseitigen Leiterplatte (Leiterseite)*



*Bild 8 Bestückungsseite der Platine*

### 3. Das NMEA-Protokoll

Für alle, die später etwas tiefer in die GPS-Welt vordringen wollen, stellen wir hier das NMEA 0183 - Protokoll (National Marine Electronics Association) vor. Dieser komplexe Standard regelt nicht nur den Datenverkehr der GPS-Geräte, sondern aller wichtigen Navigationsgeräte in der Schifffahrt (wie z.B. Radar, Wetterdaten usw.). Weitere Informationen kann man sich z.B. bei:

<http://www.nmea.de/index.html>

beschaffen. Fast alle GPS-Empfänger mit Kommunikations-Schnittstelle beherrschen zumindest diese Methode des Datenaustausches. Diese Datenfolge wollen wir am Beispiel des eTrex näher untersuchen.

Der Garmin eTrex arbeitet mit NMEA 0183 Version 2.0 und sendet die Standard-Datensätze: GPRMC, GPRMB, GPGGA, GPGSA, GPGSV, GPGLL, GPBOD und GPRTE.

Dazu kommen eigene Garmin-Datensätze:

PGRME (geschätzter Fehler), PGRMZ (Höhe). Der eTrex liefert alle 2 Sekunden einen Datenblock mit folgender Struktur (hier Standort Erfurt):

```
$GPRMC,125326,A,5057.1284,N,01102.9801,E,0.0,303.1,271002,1.0,E,A*16
$GPRMB,A,,,,,,,,,,,,,A,A*0B
$GPGGA,125326,5057.1284,N,01102.9801,E,1.04,1.9,262.5,M,46.6,M,,*42
$GPGSA,A,3,05,07,09,,,26,,,,,,,,,2.7,1.9,1.0*31
$GPGSV,2,1,08,05,28,234,50,07,43,085,33,09,63,289,45,18,25,273,00*76
$GPGSV,2,2,08,23,05,212,00,26,45,171,48,28,19,056,00,29,33,166,00*71
$GPGLL,5057.1284,N,01102.9801,E,125326,A,A*4C
$GPBOD,,T,,M,,*47
$PGRME,7.8,M,14.1,M,16.7,M*25
$PGRMZ,861,f,3*14
$GPRTE,1,1,c,*37
```

der nächste Datensatz kommt 2 Sekunden später:

```
$GPRMC,125328,A,5057.1284,N,01102.9801,E,0.0,303.1,271002,1.0,E,A*18
$GPRMB,A,,,,,,,,,,,,,A,A*0B
$GPGGA,125328,5057.1284,N,01102.9801,E,1.04,1.9,262.5,M,46.6,M,,*4C
$GPGSA,A,3,05,07,09,,,26,,,,,,,,,2.7,1.9,1.0*31
$GPGSV,2,1,08,05,28,234,50,07,43,085,33,09,63,289,45,18,25,273,00*76
$GPGSV,2,2,08,23,05,212,00,26,45,171,48,28,19,056,00,29,33,166,00*71
$GPGLL,5057.1284,N,01102.9801,E,125328,A,A*42
$GPBOD,,T,,M,,*47
$PGRME,7.8,M,14.1,M,16.7,M*25
$PGRMZ,861,f,3*14
$GPRTE,1,1,c,*37
```

Wir erkennen, dass alle „echten“ NMEA-Zeilen mit einem \$GP beginnen, eine allgemeine Kennung für GPS-Geräte. Danach folgen weitere 3 Buchstaben, die die Art der Meldung charakterisieren. Dann kommen, immer getrennt durch Komma, echte Daten. Die vorletzten beiden Zeilen (\$PG...) sind Garmin-spezifisch und weichen bereits bezüglich ihrer Kennung ab.

Untersuchen wir im folgenden einmal den oben stehenden Datensatz:

\$GPRMC,125326,A,5057.1284,N,01102.9801,E,0.0,303.1,271002,1.0,E,A\*16

**RMC - Recommended Minimum Navigation Information**

```

      1      2 3      4 5      6 7 8 9 10 11|
      |      | |      | |      | | | | | |
$--RMC,hhmmss.ss,A,llll.ll,a,yyyy.yy,a,x.x,x.x,xxxx,x.x,a*hh<CR><LF>
  
```

Field Number:

- 1) UTC Time (12:53:26)
- 2) Status, V = Navigation receiver warning, A = o.k. (A)
- 3) Latitude (5057.1284)
- 4) N or S (N)
- 5) Longitude (01102.9801)
- 6) E or W (E)
- 7) Speed over ground, (0.0) knots
- 8) Track made good, degrees true (303.1)
- 9) Date, ddmmyy (27.10.02)
- 10) Magnetic Variation, degrees (1.0)
- 11) E or W (E)
- 12) Checksum (16)

\$GPRMB,A,,,,,,,,,,,,,A,A\*0B

**RMB - Recommended Minimum Navigation Information**

```

      1 2 3 4 5 6      7 8      9 10 11 12 13|
      | | | | | |      | |      | | | | | |
$--RMB,A,x.x,a,c--c,c--c,llll.ll,a,yyyy.yy,a,x.x,x.x,x.x,A*hh<CR><LF>
  
```

Field Number:

- 1) Status, V = Navigation receiver warning
- 2) Cross Track error - nautical miles
- 3) Direction to Steer, Left or Right
- 4) TO Waypoint ID
- 5) FROM Waypoint ID
- 6) Destination Waypoint Latitude
- 7) N or S
- 8) Destination Waypoint Longitude
- 9) E or W
- 10) Range to destination in nautical miles
- 11) Bearing to destination in degrees True
- 12) Destination closing velocity in knots
- 13) Arrival Status, A = Arrival Circle Entered
- 14) Checksum (0B)

\$GPGGA,125326,5057.1284,N,01102.9801,E,1,04,1.9,262.5,M,46.6,M,,\*42

**GGA - Global Positioning System Fix Data, Time, Position and fix related data for a GPS receiver.**

```

      1      2      3 4      5 6 7 8 9 10 | 11 12 13 14 15
      |      |      | |      | | | | | | | | | |
$--GGA,hhmmss.ss,llll.ll,a,yyyy.yy,a,x,xx,x.x,x.x,M,x.x,M,x.x,xxxx*hh<CR><LF>
  
```

Field Number:

- 1) Universal Time Coordinated (UTC) (12:53:26)
- 2) Latitude (5057.1284)
- 3) N or S (North or South) (N)
- 4) Longitude (01102.9801)
- 5) E or W (East or West) (E)
- 6) GPS Quality Indicator, (1)
  - 0 - fix not available,
  - 1 - GPS fix,
  - 2 - Differential GPS fix
- 7) Number of satellites in view, 00 - 12 (04)
- 8) Horizontal Dilution of precision (1.9)

- 9) Antenna Altitude above/below mean-sea-level (geoid) (262.5)
- 10) Units of antenna altitude, meters (m)
- 11) Geoidal separation, the difference between the WGS-84 earth (46.6) ellipsoid and mean-sea-level (geoid), "-" means mean-sea-level below ellipsoid
- 12) Units of geoidal separation, meters (m)
- 13) Age of differential GPS data, time in seconds since last SC104 type 1 or 9 update, null field when DGPS is not used
- 14) Differential reference station ID, 0000-1023
- 15) Checksum (42)

\$GPGSA,A,3,05,07,09,,,26,,,,,,2.7,1.9,1.0\*31

### GSA - GPS DOP and active satellites

```
      1 2 3                14 15 16 17 18
      | | |                | | | | |
$--GSA,a,a,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x*x*hh<CR><LF>
```

Field Number:

- 1) Selection mode
- 2) Mode
- 3) ID of 1st satellite used for fix (05)
- 4) ID of 2nd satellite used for fix (07)
- ...
- 14) ID of 12th satellite used for fix
- 15) PDOP in meters (2.7)
- 16) HDOP in meters (1.9)
- 17) VDOP in meters (1.0)
- 18) checksum (31)

\$GPGSV,2,1,08,05,28,234,50,07,43,085,33,09,63,289,45,18,25,273,00\*76

\$GPGSV,2,2,08,23,05,212,00,26,45,171,48,28,19,056,00,29,33,166,00\*71

### GSV - Satellites in view

```
      1 2 3 4 5 6 7      n
      | | | | | | |      |
$--GSV,x,x,x,x,x,x,x,...*hh<CR><LF>
```

Field Number:

- 1) total number of messages (2)
- 2) message number (1 oder 2)
- 3) satellites in view (08)
- 4) satellite number (05) 1.Sattellit
- 5) elevation in degrees (28)
- 6) azimuth in degrees to true (234)
- 7) SNR in dB (50)
- more satellite infos like 4)-7) (07,43,085,33 usw.)
- n) checksum (76 oder 71)

\$GPGLL,5057.1284,N,01102.9801,E,125328,A,A\*42

### GLL - Geographic Position - Latitude/Longitude

```
      1      2 3      4 5      6 7
      |      | |      | |      | |
$--GLL,llll.ll,a,yyyyy.yy,a,hmmss.ss,A*hh<CR><LF>
```

Field Number:

- 1) Latitude (5057.1284)
- 2) N or S (North or South) (N)
- 3) Longitude (01102.9801)
- 4) E or W (East or West) (E)
- 5) Universal Time Coordinated (UTC) (12:53:28)
- 6) Status A - Data Valid, V - Data Invalid
- 7) Checksum (42)

\$GPBOD,,T,,M,,\*47

### BOD - Bearing - Waypoint to Waypoint

```
      1 2 3 4 5 6 7
      | | | | | | |
$--BOD,x.x,T,x.x,M,c--c,c--c*hh<CR><LF>
```

Field Number:

- 1) Bearing Degrees, TRUE
- 2) T = True (T)
- 3) Bearing Degrees, Magnetic
- 4) M = Magnetic (M)
- 5) TO Waypoint
- 6) FROM Waypoint
- 7) Checksum (47)

\$PGRME,7.8,M,14.1,M,16.7,M\*25

### RME - Estimated Position Error

Field Number:

- 1) Estimated horizontal position error (HPE)(7.8)
- 2) Unit, meters (m)
- 3) Estimated vertical position error (HVE) (14.1)
- 4) Unit, meters (m)
- 5) Overall spherical equivalent position error (16.7)
- 6) Unit, meters (m)
- 7) Checksum (25)

\$PGRMZ,861,f,3\*14

### RMZ - Altitude information

Field Number:

- 1) Altitude (861)
- 2) Unit (f, feets)
- 3) Position fix dimensions: 2=user / 3=GPS
- 7) Checksum (14)

\$GPRTE,1,1,c,\*37

### RTE - Routes

```
      1 2 3 4 5      x  n
      | | | | |      |  |
$--RTE,x.x,x.x,a,c--c,c--c, ..... c--c*hh<CR><LF>
```

Field Number:

- 1) Total number of messages being transmitted (1)
- 2) Message Number (1)
- 3) Message mode (c)  
c = complete route, all waypoints  
w = working route, the waypoint you just left, the waypoint you're heading to then all the rest
- 4) Waypoint ID
- x) More Waypoints
- n) Checksum (37)

## 4. Der Text-Mode des eTrex

Wenn man schon einen Garmin eTrex hat, ist vielleicht auch der viel einfachere Text-Mode für die Datenübertragung und -auswertung interessant. Hier ist die Datenfülle deutlich reduziert, alle Datensätze haben die gleiche Länge und jeder Datensatz hat dieselbe Struktur. Das ist ideal für die Auswertung mit einem Steuercomputer wie dem BASIC-Tiger® (sehr einfache Stringbearbeitung). Im folgenden auch dazu ein Beispiel:



@000607204655N6012249E01107556S015+00130E0021N0018U0000

@yymmddhhmmss Latitude Longitude error Altitude EWSpd NSSpd VSpd

Jede Eintragung hat eine fixe Länge, die nur durch ihre Position im String gekennzeichnet ist. Bemerkenswert ist, dass bei aller Kürze neben der Position (X, Y, Z) auch die Geschwindigkeit in X- Y- und Z-Richtung ausgegeben wird – Ballon-Fahrer werden das zu schätzen wissen! Der Datensatz wird wie folgt interpretiert:

@000607204655N6012249E01107556S015+00130E0021N0018U0000

@		jeder Datensatz beginnt mit @
000607		UTC-Datum (07.06.2000)
204655		Uhrzeit (20:46:55)
N6012249		Breite (N6012.249)
E01107556		Länge (E01107.556)
S		Positions-Status* (S)
015		Horiz. Pos. Fehler (015)
+		Vorzeichen Höhe (+)
00130		Höhe in m (00130)
E		Geschw. Richtg. Ost (E)
0021		Betrag/10 (m/s)(002.1)
N		Geschw. Richtg. Nord (N)
0018		Betrag/10 (m/s) (001.8)
U		Geschw. Richtg. Oben (U)
0000		Betrag/100 (m/s)(00.00)

Position Breite:  
Immer Dezimalzeichen nach 4 Stellen  
Position Länge:  
Immer Dezimalzeichen nach 5 Stellen  
mögliche Zeichen Position und Geschwindigkeit:  
N, S, E, W (Nord, Süd, Ost, West)  
Mögliche Zeichen Höhe:  
+, - (über/unter Meeresspiegel)  
Mögliche Zeichen vertikale Bewegung:  
U, D (aufwärts, abwärts)  
Positions Status:  
d (2D differential GPS Position)  
D (3D differential GPS Position)  
g (2D GPS Position)  
G (3D GPS Position)  
S (simulierte Position)  
- (ungültige Position)

Alle Werte bekommen automatisch führende Nullen, bis die vorgeschriebene Länge erreicht ist!

## 5. Funktionsweise des Programms

Das Programm „GPS-REC.TIG“ im BASIC-Tiger<sup>®</sup> verfügt über drei Tasks: Den Haupt-Task „Main“, in dem zunächst die Treiber initialisiert werden und dann in einer Schleife der Port 8 abgefragt wird, und zwei weitere Tasks in denen die empfangenen Zeichen der beiden seriellen Schnittstellen SER0 und SER1 zu Strings zusammengefügt werden.

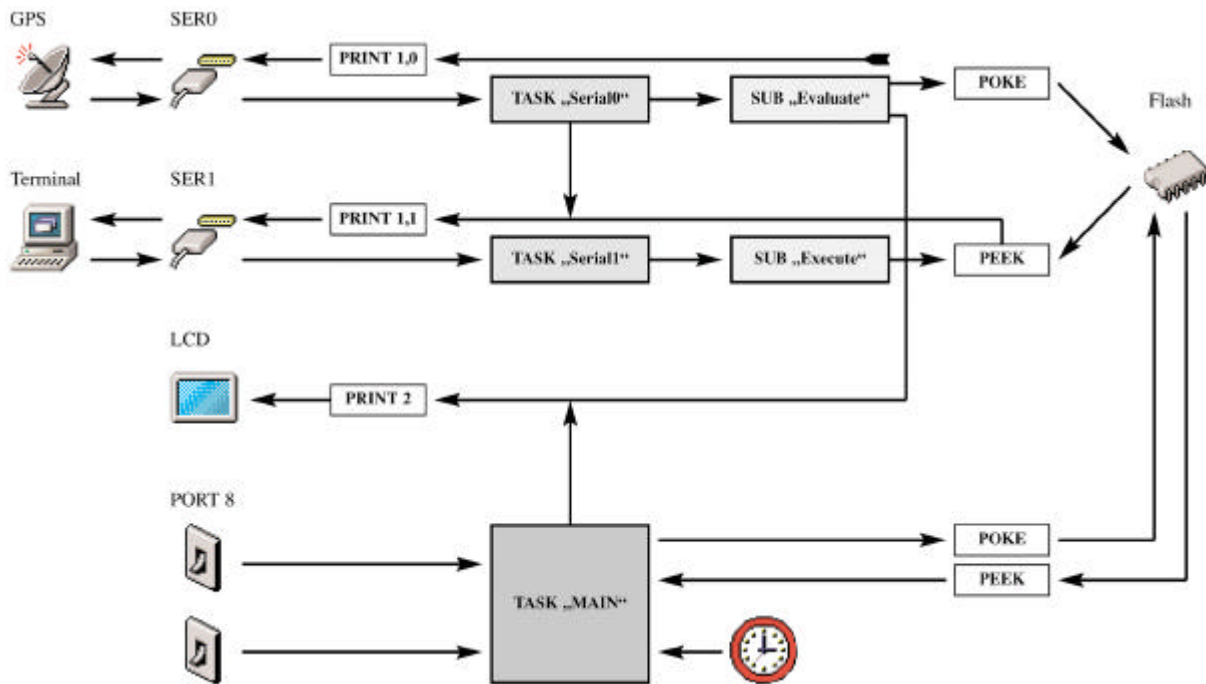


Bild 9 Schematischer Programmablauf

Wenn im Task „Serial0“ ein Steuerzeichen empfangen wird, so wird dies als Satzende erkannt. Der bis dahin gespeicherte String enthält einen NMEA Datensatz vom GPS Empfänger, der in der Subroutine „Evaluate“ weiterverarbeitet wird. Zusätzlich wird das empfangene Telegramm über die andere serielle Schnittstelle wieder gesendet, falls das Programm sich im „NMEA GPS“ Modus befindet. In diesem Modus kann ein an der Schnittstelle „SER1“ angeschlossenes Terminal die Daten so empfangen als ob sie direkt vom Empfänger kämen.

Der Task „Serial1“ verhält sich ähnlich. Hier werden die Zeichen gesammelt, die vom angeschlossenen Terminal an der Schnittstelle „SER1“ gesendet werden. Ist ein Satz vollständig, so wird er als Befehl in der Subroutine „Execute“ weiterverarbeitet. Auf jeden gültigen Befehl wird eine Antwort auf derselben Schnittstelle zurückgesendet. Um eine ungünstige Durchmischung mit den NMEA Telegrammen vom GPS Empfänger zu vermeiden, gibt es die zwei Betriebs-Modi: Im „NMEA GPS“-Modus werden, wie oben beschrieben, die Telegramme vom Empfänger durchgereicht. Im „Command“-Modus werden die Befehle vom Terminal verarbeitet und die Antworten zurückgegeben.

Die Umschaltung der Betriebsmodi geschieht durch Öffnen oder Schließen des Schalters an Pin 10 (Port 8.1). Zusätzlich kann das Ende einer Fahrt (Abschalten der Zündung) festgestellt werden, wenn der 1F Super-Cap in die Schaltung eingebaut wird. In diesem Fall wird das Signal an Pin 21 (Port 8.5) vom BASIC-Tiger® ausgewertet. Fehlt der Kondensator, so wird nicht das Ende einer Fahrt, sondern der Anfang einer Fahrt gespeichert.

Die Hauptaufgabe des Programms im BASIC-Tiger® ist es den Anfang oder das Ende einer Fahrt sowie viele Zwischenschritte im Flash-Speicher des Tigers abzulegen. Befindet sich das

Programm im Kommandomodus, so können diese Daten über die serielle Schnittstelle später wieder ausgelesen werden.

## 5.1. Speicherblöcke

Das Abspeichern und Auslesen von Flash-Speicher geschieht in Datensätzen, die in einem Datenkopf und einem Datenrumpf unterteilt sind. Der Datenkopf besteht immer aus einem einzelnen Byte und dient der Kennzeichnung der nachfolgenden Daten.

## 5.2. Datentypen

Um den Datentyp zu bestimmen, werden immer nur die ersten 3 Bit des Datenkopfes ausgewertet:

Typ	Bedeutung	Folge-Bytes
0	Fahrtende	18 Bytes
1	Fahrtanfang	14 Bytes
2	Schritt kompr.	1 Byte
3	Schritt kurz	3 Bytes
4	Schritt lang	6 Bytes
5	Schritt voll	12 Bytes
6	DEBUG	4 Bytes
7	Nicht definiert	---

Ist im Datenkopf das Bit 7 gesetzt (also irgendein Wert  $> 127$ ) so bedeutet dies einen ungültigen Wert. Das Programm erkennt damit den nicht benutzen Flash-Speicher da beim Löschen alle Bytes mit 11111111b belegt sind. Die Folge-Bytes im Datenrumpf dürfen beliebige Werte haben.

## 5.3. Dateninhalt

Als Dateninhalt werden Positionen, Datum/Uhrzeit, Fahrtzeit und Fahrstrecke gespeichert. Eine Position besteht aus 2 LONG die die Latitude (Breite) und die Longitude (Länge) als Dezimalgrad mal 10000 darstellen. Die Angaben enthalten also den Wert in Grad sowie 4 dezimale Nachkommastellen.

Datum und Uhrzeit werden in einem LONG als die Sekunden seit dem 1.1.1980 0:00 Uhr dargestellt. Diese Formatierung ist weit verbreitet und bringt bei Berechnungen einige Vorteile. Um diesen Wert z.B. in die Microsoft Windows konforme Fließkommazahl für die Zeitdarstellung umzuwandeln, kann folgende Formel verwendet werden:  $(\text{SEKUNDEN} / 86400) + 29221$ .

Die Fahrtzeit wird als WORD gespeichert und enthält die Sekunden seit Fahrtbeginn. Die Zeit darf also 18 Stunden nicht überschreiten.

---

Die Fahrtstrecke wird in einem WORD gespeichert und enthält die gefahrenen Hektometer (Kilometer \* 10). Der Maximalwert liegt also bei 6553 Kilometer pro Fahrt.

Die Datentypen 0 und 1 speichern zusätzlich noch den so genannten TTFW-Wert (Time to first fix) ab. Dieser Wert gibt an, wie lang der Empfang der ersten gültigen Koordinate nach Fahrtbeginn gedauert hat.

Die Datentypen 0,1 und 5 enthalten die Absolutwerte wogegen die Typen 2,3 und 4 Differenzwerte zum vorigen Wert enthalten, um möglichst kurz gespeichert zu werden. Besonders erwähnenswert ist hierbei der Datentyp 2. Dieser ermöglicht das Abspeichern eines Koordinatenwechsels mit Breite und Länge sowie der gefahrenen Sekunden in einem einzigen Byte!

#### **5.4. Datentyp 2**

Die Speicherung als Datentyp 2 hat folgende Voraussetzungen: Der Koordinatenwechsel ist nicht größer als +/- 1 bei Breite und Länge und die vergangenen Sekunden betragen nicht mehr als 31. Die Speicherung des Koordinatenwechsels benötigt hierbei tatsächlich nur 3 Bit, die restlichen 5 Bit stehen der Zeitspeicherung zur Verfügung. Der Koordinatenwechsel wird dabei folgendermaßen kodiert: Die Differenz zur vorherigen Koordinate kann immer in einer 3 mal 3 Matrix gezeichnet werden wobei die vorige Koordinate immer der Mittelpunkt der Matrix ist. Da von den 9 Feldern der Matrix als mögliche Richtungswechsel somit 8 Felder verbleiben, können diese 8 möglichen Richtungswechsel in 3 Bit kodiert werden. Die Richtungswechsel von der vorigen zur aktuellen Koordinate sind folgendermaßen definiert:

0	Nord
1	Nord-Ost
2	Ost
3	Süd-Ost
4	Süd
5	Süd-West
6	West
7	Nord-West

#### **5.5. Datentyp 3**

Die Speicherung als Datentyp 3 hat folgende Voraussetzungen: Der Koordinatenwechsel ist nicht größer als +/- 255 bei Breite und Länge und die vergangenen Sekunden betragen nicht mehr als 63. Im ersten Byte wird der Absolutbetrag der Breitendifferenz gespeichert. Im zweiten Byte wird der Absolutbetrag der Längendifferenz gespeichert. Im dritten Byte wird in den Bits 0-5 die Zeitdifferenz gespeichert. Im Bit 7 wird das Vorzeichen der Breitendifferenz gespeichert. Im Bit 6 wird das Vorzeichen der Längendifferenz gespeichert.

## 5.6. Datentyp 4

Die Speicherung als Datentyp 4 hat folgende Voraussetzungen: Der Koordinatenwechsel ist nicht größer als +/- 32767 bei Breite und Länge. Im ersten Wort wird die Breitendifferenz plus 32767 gespeichert. Im zweiten Wort wird die Längendifferenz plus 32767 gespeichert. Im dritten Wort wird die Zeitdifferenz gespeichert.

## 5.7. Kommandos

Um eine einfache Befehlsstruktur zu implementieren, wertet die Subroutine „Execute“ immer nur das erste Zeichen eines Befehls aus und führt dann die entsprechenden Befehle aus. Alle Antworten für diese Befehle beginnen mit einem Klammeraffen (@), so dass die Auswertesoftware leicht erkennen kann, ob eine Antwort vorliegt.

Bei den Befehlen „?“ „“ „:“ „;“ wird am Ende der Antwort ein Stern gefolgt von einer Prüfsumme in Hexadezimalkodierung angehängt. Die Prüfsummenberechnung entspricht dem Verfahren nach NMEA, so dass hier die gleiche Prüfroutine verwendet werden kann.

Schlägt die Prüfung fehl, so wird beim Befehl „?“ die gleiche Abfrage wiederholt. Bei den Punkt-Befehlen („“ „:“ „;“) wird die vorige Abfrage mit dem Befehl „;“ wiederholt.

Die Punkt-Befehle („“ „:“ „;“) geben als letzten Parameter vor der Prüfsumme den Speicherplatz der Daten aus. Wird diese Angabe bei den Befehlen („“ oder „:“) verwendet, so kann ein Download nach und nach fortgesetzt werden.

### 5.7.1. Versionsabfrage ?

Abfrage von Version, Anzahl Datensätze, Anzahl Fahrten, Speicheradresse, Gesamtspeicher. Die Lese-Adresse für den Punkt-Befehl („“ oder „:“) wird zurückgesetzt.

[@3.1,573,7,2219,393216\\*2E](#)

Die einzelnen Parameter zwischen „@“ und „\*” sind durch Kommas voneinander abgetrennt.

Folgende Parameter werden ausgegeben:

- Version Firmware
- Gespeicherte Datensätze
- Davon gespeicherte Fahrten
- Benutzter Flash-Speicher
- Gesamtgröße des Flash-Speichers

### 5.7.2. Datensatz lesen .{Adresse}

Nächsten Datensatz auslesen. Wird hinter dem Punkt eine Zahl angegeben, so wird der Datensatz an dieser Adresse gelesen (optional).

[@2,21,91\\*39](#)

Die einzelnen Parameter zwischen „@“ und „\*” sind durch Kommas voneinander abgetrennt.

Je nach Datensatztyp werden folgende Daten ausgegeben:



- 0 Breite, Länge, Uhrzeit, Strecke, Fahrtzeit, TTFF
- 1 Breite, Länge, Uhrzeit, TTFF
- 2 Delta Breite + Delta Länge + Delta Fahrtzeit
- 3 Delta Breite, Delta Länge, Delta Fahrtzeit
- 4 Delta Breite, Delta Länge, Delta Fahrtzeit
- 5 Breite, Länge, Uhrzeit
- 6 Beliebiger LONG Zahlenwert

Sind keine weiteren Daten vorhanden, so erfolgt die Ausgabe

@EOD\*4E

### 5.7.3. Datensatz lesen :{Adresse}

Nächsten Datensatz auslesen. Im Gegensatz zum Befehl „,“ wird nur das nächste Fahrtende übermittelt. Wird hinter dem Punkt eine Zahl angegeben, so wird der Datensatz an dieser Adresse gelesen (optional).

@0,539414,103073,739738056,184,1370,8,1131\*24

Die einzelnen Parameter zwischen „,@“ und „,\*“ sind durch Kommas voneinander abgetrennt. Je nach Datensatztyp werden folgende Daten ausgegeben:

- 0 Breite, Länge, Uhrzeit, Strecke, Fahrtzeit, TTFF
- 1 Breite, Länge, Uhrzeit, TTFF

Sind keine weiteren Daten vorhanden, so erfolgt die Ausgabe

@EOD\*4E

### 5.7.4. Abfrage wiederholen ;

Letzten Punkt-Befehl („,“ oder „:“) wiederholen. Sinnvoll wenn Prüfsumme falsch.

@0,539414,103073,739738056,184,1370,8,1131\*24

### 5.7.5. Datensatz speichern ,

Datensatz speichern. Ein zuvor aufgezeichneter Datensatz kann zurückgeschrieben werden. Dazu wird der Satz, so wie er empfangen wurde (jedoch ohne „,@“) plus einem vorangestellten und zwei nachgestellten Kommas angegeben.

@Next Flash address: xxxx

### 5.7.6. An Empfänger \$

NMEA Datensatz an Empfänger. Der Datensatz wird unverändert an den Empfänger weitergegeben.

### 5.7.7. Speicher löschen !

Flash Speicher löschen. Es erfolgt für jede gelöschte Speicherbank eine Rückmeldung

... @FLASH ERASED!

### 5.7.8. Gerätestatus abfragen -

Mit diesem Befehl kann getestet werden, ob das Gerät angeschlossen ist und sich im Kommandomodus befindet.

@OK

### 5.8. Details zum BASIC-Tiger® Programm

Der Quellcode „GPS-REC.TIG“ enthält folgende Highlights:

- Zeitlich unabhängiger Empfang von Strings auf der seriellen Schnittstelle in einem separaten Task --> „TASK Serial0“ und „TASK Serial1“
- Umschalten der Baudrate zur Laufzeit --> „SUB CheckInputs“
- Berechnung der Entfernung zwischen zwei Koordinaten --> „SUB GetDistance“
- Effiziente Umrechnung von Datum/Uhrzeit in Sekunden seit 1.1.1980 --> „SUB GpsTime“. Die vorgestellte Routine ist kompakter als das original Wilke-Beispiel in der Datei „Timecvt.tig“
- Errechnen einer Prüfsumme nach NMEA Standard --> „SUB AddNmea“
- Prüfen einer Prüfsumme nach NmeaStandard --> „SUB NmeaCheck“
- Kodierung eines Koordinatenwechsels in nur 3 Bit --> „SUB WriteFlash“

Wir hoffen, Ihnen das interessante Gebiet des GPS ein wenig näher gebracht zu haben. Die vorgestellte einfache Anwendung als Fahrten-Tagebuch lässt sich in vielerlei Richtung variieren und erweitern. Eine „Blackbox-Version“ dieses Fahrtenbuchs finden Sie unter:

<http://www.dynamo-software.de>

Stellt man sich vor, die gewünschten Daten nicht im Flash des Tigers, sondern in einer SmartMedia-Karte zu speichern, kann eine viel größere Datenmenge erfasst werden. Außerdem ist die Auswertung einfacher – nach der Fahrt SM-Karte raus und in den Kartenleser am PC rein. Sofort können alle Einzelheiten der Autofahrt, Wanderung oder Ballonfahrt im Detail untersucht werden. Weitere Anwendungen sind denkbar – der GPS-Empfänger in ihrem Fahrzeug gibt Daten an den BASIC-Tiger® weiter, wenn das Fahrzeug z.B. unbefugt benutzt wird. Dieser macht daraus eine SMS für ein Handy, die dann immer den augenblicklichen Standort des Autos an Sie oder eine Wachfirma abgibt... Tolle Perspektiven, nicht wahr?