

Minimalistischer Selbstbau-GPS-Logger ohne μ Controller und SD-Karte Gunther Zielosko

GPS-Logger für Radfahrer, Wanderer, Autofahrer und andere gibt es in großer Auswahl zu kaufen, leider meist nicht billig. Ebenso viele Selbstbau-Varianten werden in diversen Internet-Foren angeboten. Wenn manche auch schon sehr einfach nachzubauen sind, einige handwerkliche Fähigkeiten und/oder Programmierkenntnisse werden fast immer vorausgesetzt. Das absolute „Muss“ scheinen dabei immer Microcontroller, etwa ATMEGA und eine SD-Karte zu sein. Doch es geht auch anders - hier nun etwas für den puristischen Bastler alter Schule, der nur Löten kann...

Der GPS-Baustein L80 von Quectel

Auch dieser Baustein ist minimalistisch – allerdings nur von der Größe her – 16 mm x 16 mm x 6,45mm. Das Gehäuse mit integrierter Antenne hat 12-pin 2.54mm pitch SMT Anschlüsse, die entweder direkt ohne Draht auf typische durchkontaktierte Experimental-Leiterplatten mit gleichem Rastermaß gelötet oder mit angelöteten Pfostensteckern in eine entsprechende Fassung gesteckt werden können. Trotz seiner geringen Größe ist das GPS-Modul extrem empfindlich und leistungsfähig. Technische Daten, Applikationshinweise und anderes können hier eingesehen werden:

<http://www.quectel.com/product/prodetail.aspx?id=62>

http://www.quectel.com/uploadimage/download/l80_hardware_design_v1.1.pdf

http://www.quectel.com/UploadFile/Product/Quectel_L80_GPS_Specification_V1.1.pdf

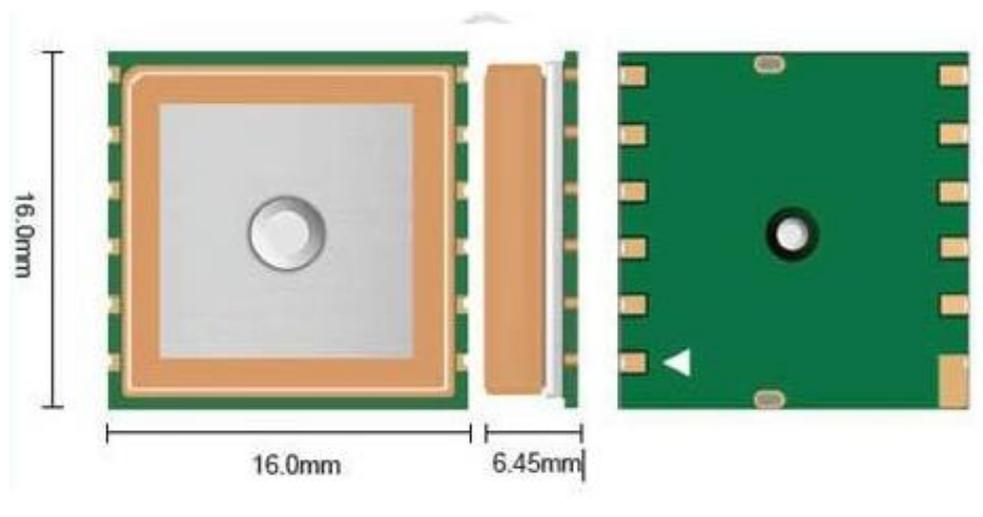


Bild 1 Der Baustein L80 von Quectel

Die in ersten Versuchen gewonnenen Ergebnisse sind beeindruckend – sogar weit im Zimmer gab es regelmäßig einen Empfang und einen sicheren Fix. Doch dazu später.

Der Baustein arbeitet bei nominell 3,3 V (3,0...4,3 V). Zunächst stand die Frage, wie kommuniziert der L80 mit der Außenwelt. Wie bei aktuellen Modellen üblich, sendet er seine Daten über die (nur scheinbar veraltete!) serielle Schnittstelle, allerdings auch wie üblich bei GPS-Empfängern im TTL-Format. Da der Baustein für einen Betrieb mit nominell 3,3 V (typisch Li-Ion oder 3 Zellen NiMH) vorgesehen ist, liegen auch die Ausgangs-Pegel bei 0V bzw. 3,3 V. Das heißt, um echte RS232-Signale zu generieren, genügt hier kein MAX232. Es muss ein MAX3232 oder vergleichbarer Typ verwendet werden, der mit einer Betriebsspannung von 3,3 V die geforderten Pegel für RS232 liefern kann. Für den einfachsten Betrieb – Spannung anlegen und am RS232-Ausgang die typischen GPS-Daten im NMEA-Format auszugeben, reicht die folgende Schaltung schon aus:

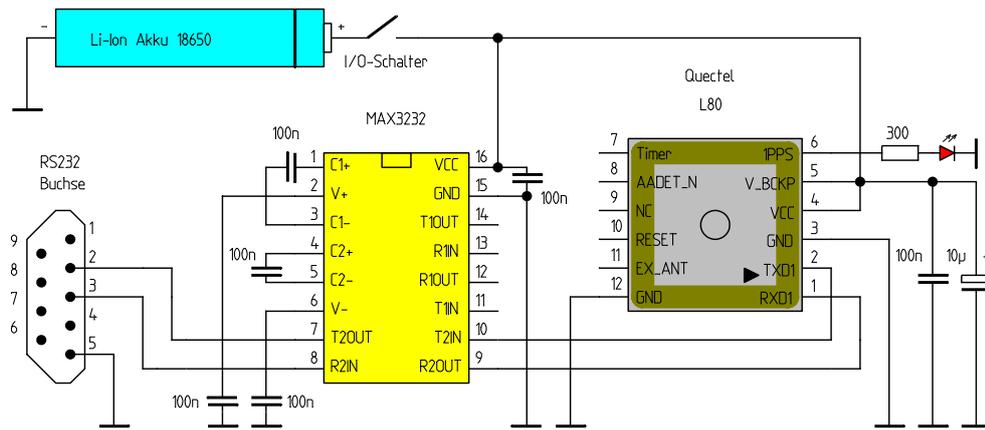


Bild 2 Die komplette Schaltung zum Übertragen der NMEA-Daten und Steuersignalen

Profi-Bastler werden nun sagen – Na und? So etwas ist mit fast jedem beliebigen GPS-Modul einfach zu bewerkstelligen. Und – das ist noch lange kein GPS-Logger. Stimmt soweit, aber jetzt kommt eine besondere Fähigkeit des L80 zu Einsatz. Er hat nämlich einen umfangreichen inneren Speicher zum Ablegen der empfangenen und von ihm selbst „ausgerechneten“ Daten. Zwar werden diese in etwas kryptischer Form gespeichert, sind aber über die RS232-Schnittstelle auszulesen. Auch Steuerkommandos können an das Modul gesendet werden. Ein einfaches, aber wirkungsvolles Tool für den PC unter Windows (funktioniert bis Windows 10/64Bit!) stellte die Firma Quectel nach Anfrage zur Verfügung (hier angehängt):



Quectel L76 LOCUS manager V1.0a

Das Tool benötigt für den Dialog mit dem GPS-Logger allerdings eine echte RS232-Schnittstelle. Wenn der PC keine mehr hat, geht es auch mit einem geeigneten USB/Seriell-Wandler. Als sehr geeignet im Gegensatz zu vielen anderen auch für Windows10/64 Bit hat sich dieser erwiesen:

DIGITUS DA 70156 RS 232 USB Seriell DB9 ADAPTER Serial Stick Dongle Null Modem

z.B. von:

<https://www.amazon.de/Digitus-USB-Seriell-Adapter-schwarz/dp/B0030IT780>



Bild 3 der verwendete USB/Seriell-Wandler DIGITUS DA 70156

Wenden wir uns nun dem LOCUS Manager zu. Es ist nur eine EXE-Datei, die ohne Installation ausführbar ist. Damit etwas passiert, schließt man den eingeschalteten GPS-Logger an eine RS232-

Schnittstelle des PC an. Wenn keine vorhanden ist, wird der o.a. USB/Seriell-Wandler benutzt. Im Geräte-Manager findet man den entsprechenden COM-Anschluss, der dann für den GPS-Logger benutzt werden kann. Nach dem Starten des LOTUS-Managers erwartet dieser, dass man als erstes diese COM-Schnittstelle für die Verbindung angibt. Die Baudrate ist mit 9600 Bd schon voreingestellt und passt erst einmal für den unveränderten L80.

Wenn alles funktioniert, sollte der LOCUS-Manager jetzt bereits anzeigen, dass NMEA-Daten vorliegen (Bild 4). Als nächstes klickt man mit dem Mauszeiger im Commandfeld rechts auf den Pfeil und es werden verschiedene Kommandos für den L80 angeboten (Bild 5). Wählt man „Request LOCUS status“ aus, sieht man, was derzeit im L80 eingestellt ist (Bild 6). Vielleicht beim ersten Einsatz, dass LOCUS nicht eingeschaltet ist (LOCUS OFF), voreingestellt ein Intervall von 15 Sekunden programmiert, der Betriebs-Modus fullstop und nichts im Speicher steht (total 0, 0%).

Die im Schaltplan eingezeichnete LED ist optional, aber sinnvoll. Sie blinkt im 1 Sekundentakt, wenn der GPS-Baustein gültige Daten sendet, also im Fix ist. Zusätzlich signalisiert sie natürlich auch, dass das Gerät eingeschaltet ist.

Kommen wir jetzt zu den einzelnen Schritten, die mit dem LOCUS-Commander für den Loggerbetrieb erforderlich sind:

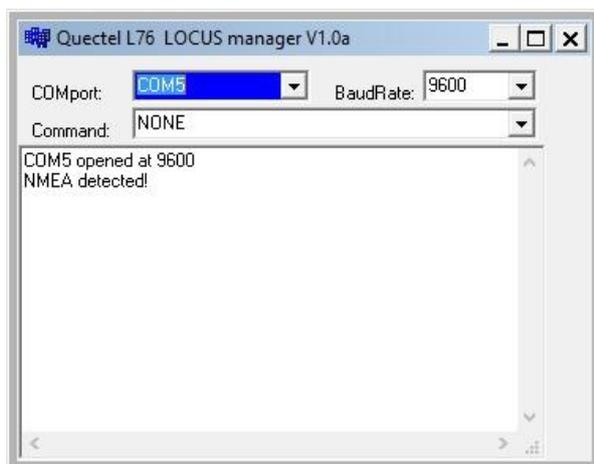


Bild 4

- Starten von locusmgr.exe
- Auswählen des COM-Ports und der Baudrate (voreingestellt 9600Bd)
- Anzeige, wenn Gerät angeschlossen (NMEA detected)

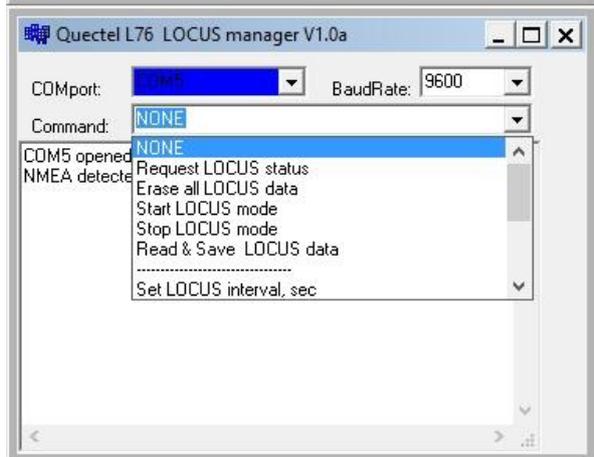


Bild 5

- Beim Klicken mit der Maus auf den Command-Bereich (Pfeil rechts) erscheinen die vorgefertigten Kommandos.
- Request LOCUS status zeigt den aktuellen Status des Systems an

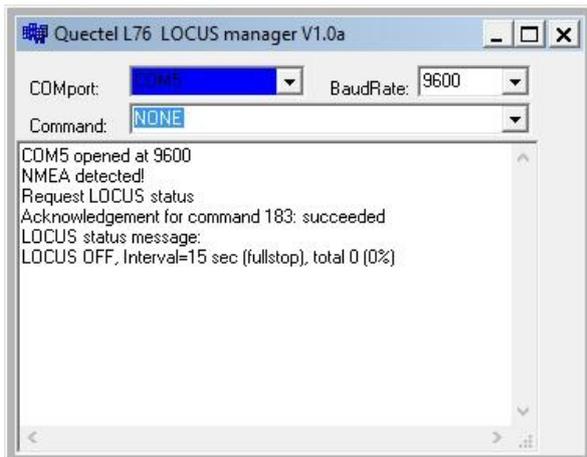


Bild 6

- LOCUS ist noch OFF, 0 (0%) Punkte gespeichert
- Voreinstellung aus 15 Sekunden Intervall

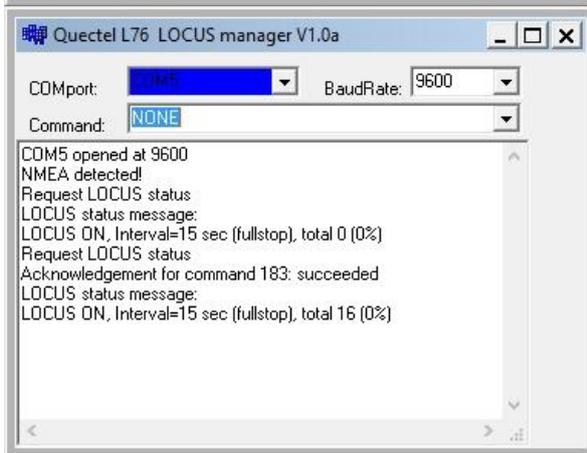


Bild 7

- LOCUS ist jetzt ON und hat 16 Datensätze gespeichert

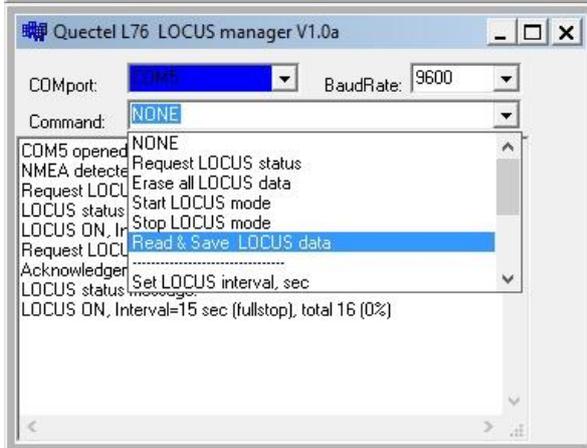


Bild 8

- Auswählen von Read & Save LOCUS data
 - Auswählen von Ziel und Namen der zu speichernden Datei (.txt ist voreingestellt, aber man kann auch z.B. .nmea eintragen, das ermöglicht ein direktes Einlesen z.B. in GoogleEarth)
- Antwort:
- Finded 18 valid LOCUS points
 - Acknowledgement for command 622: succeeded

Mit dem Kommando „Start LOCUS mode“ beginnt das Sammeln von NMEA-Daten in den internen Speicher des L80. Es werden im Standard-Modus nur die \$GPRMC-Daten gespeichert. Jetzt ist das Modul aktiviert, kann von der Schnittstelle einfach abgezogen werden und man kann mit dem Logger nun die Gegend unsicher machen.

Am Ende der Aufzeichnung, wenn man wieder zuhause ist, steckt man den Logger wieder an den PC, startet den LOCUS-Manager und wählt diesmal den Befehl „Stop LOCUS mode“. Zum Auslesen der Daten folgt der Befehl „Read & Save LOCUS data“. Im aufgehenden Dialogfeld wählt man den Namen und den Speicherort für die Daten aus. Als Dateinamen ist zunächst die Endung .txt vorgesehen, diese kann aber problemlos z.B. gleich in .nmea eingegeben werden, was ein Einlesen in GoogleEarth sehr vereinfacht.

Das Ergebnis ist eine minimalistische NMEA-Datei nur mit \$GPRMC-Sequenzen, die alle wesentlichen Standort-Daten enthält. Das Zeitintervall ist mit 15 Sekunden voreingestellt, das reicht allemal für Fußmärsche oder Radtouren. Bemerkenswert ist, dass der Speicher des L80 in dieser Einstellung für

16 Stunden reicht. Im Folgenden eine Beispieldatei, die ursprünglichen Koordinaten wurden mit einem „x“ unkenntlich gemacht:

```
$GPRMC,122530.000,A,5x58.3250,N,01x02.5815,E,0.000,0.00,010217,,A*77
$GPRMC,122545.000,A,5x58.2747,N,01x02.5999,E,0.000,0.00,010217,,A*72
$GPRMC,122600.000,A,5x58.2202,N,01x02.6234,E,0.000,0.00,010217,,A*7B
$GPRMC,122615.000,A,5x58.1953,N,01x02.6341,E,0.000,0.00,010217,,A*70
$GPRMC,122630.000,A,5x58.1925,N,01x02.6362,E,0.000,0.00,010217,,A*77
$GPRMC,122645.000,A,5x58.1918,N,01x02.6366,E,0.000,0.00,010217,,A*7F
$GPRMC,122700.000,A,5x58.1911,N,01x02.6333,E,0.000,0.00,010217,,A*76
$GPRMC,122715.000,A,5x58.1923,N,01x02.6192,E,0.000,0.00,010217,,A*7A
$GPRMC,122730.000,A,5x58.1948,N,01x02.6089,E,0.000,0.00,010217,,A*7B
$GPRMC,122818.000,A,5x58.1927,N,01x02.5986,E,0.000,0.00,010217,,A*72
$GPRMC,122833.000,A,5x58.1902,N,01x02.6026,E,0.000,0.00,010217,,A*7C
$GPRMC,122848.000,A,5x58.1861,N,01x02.6100,E,0.000,0.00,010217,,A*71
$GPRMC,122903.000,A,5x58.1836,N,01x02.6178,E,0.000,0.00,010217,,A*72
$GPRMC,122918.000,A,5x58.1863,N,01x02.6125,E,0.000,0.00,010217,,A*70
$GPRMC,122933.000,A,5x58.1877,N,01x02.6093,E,0.000,0.00,010217,,A*70
$GPRMC,122948.000,A,5x58.1875,N,01x02.6117,E,0.000,0.00,010217,,A*73
$GPRMC,123003.000,A,5x58.1868,N,01x02.6172,E,0.000,0.00,010217,,A*7B
$GPRMC,123018.000,A,5x58.1857,N,01x02.6224,E,0.000,0.00,010217,,A*7D
```

Anders als in diesem Beispiel gibt es unter dem LOCUS-Manager weitere Einstellmöglichkeiten für das Zeit-Intervall, hier sind 5-600 Sekunden einstellbar.

Darstellung in GoogleEarth

GoogleEarth wird gestartet.

Unter

Tools → GPS → Importieren → Aus Datei importieren → Importieren

wählt man die gespeicherte Datei aus und sofort erscheint die gewanderte oder geradelte Strecke als farbige Linie im bereits gezoomten Gebiet.

Handhabung in der Praxis

Wird das Gerät eingesetzt, sind folgende Abläufe einzuhalten:

Einschalten der Betriebsspannung am Logger

Verbinden des Loggers mit dem PC, ggf. über USB/Seriell-Wandler

Starten „locusmgr.exe“ auf dem PC

Einstellen der COM-Schnittstelle

Kommando „Erase all LOCUS data“ *falls noch alte Daten vorhanden*

Kommando „Start LOCUS mode“

Logger kann jetzt vom PC getrennt werden

.

Wandern, Radfahren

.

Verbinden des Loggers mit dem PC, ggf. über USB/Seriell-Wandler

Starten „locusmgr.exe“ auf dem PC

Einstellen der COM-Schnittstelle

Kommando „Stop LOCUS mode“
 Kommando „Read & Save LOCUS data“
 Eingeben von Dateiname und Speicherort
 Kommando „Erase all LOCUS data“ *wenn die Daten sicher auf dem PC sind*
 Ausschalten der Betriebsspannung am Logger

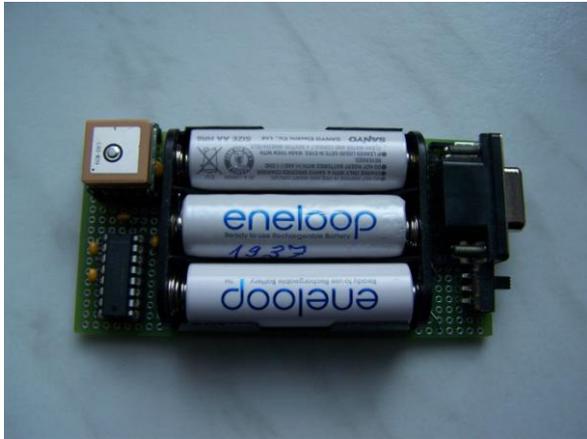


Bild 9 Einfacher Aufbau...

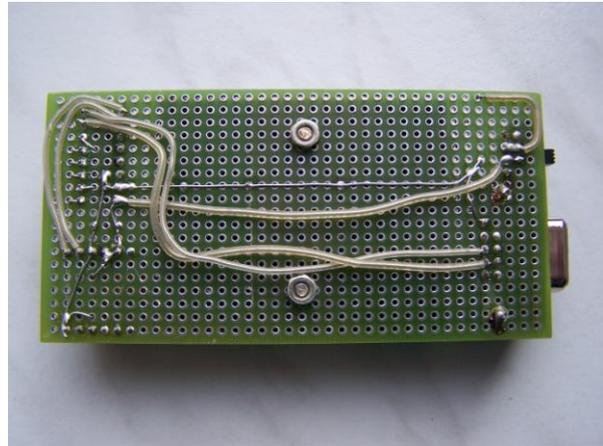


Bild 10 ... und minimaler Verdrahtungsaufwand

Es geht noch „handlicher“

Für noch unscheinbarere Anwendungen, z.B. als Hundehalsband, für die Jackentasche oder andere, lässt sich der eigentliche Logger-Aufbau noch deutlich kleiner gestalten, wenn man die gesamte Kommunikationstechnik nach außen verlegt. D.h. auf dem Logger befinden sich nur noch der L80, eine Batterie, ggf. die LED und eine kleine Pfosten-Buchse anstelle der großen RS232-Buchse. Der Baustein MAX3232 und die RS232-Buchse befinden sich auf einer Extra-Platine und diese wird nur (zusammen mit dem USB/RS232-Wandler und dem PC) zum Starten und Auslesen des Systems gebraucht. Nach dem Starten läuft der L80 völlig autark bis der Speicher voll ist oder das Gerät wieder am PC ausgelesen wird.

Und es gibt noch einen Vorteil dieser extra einfachen Variante – die Kosten des eigentliche Loggers für draußen sind nun minimal. Solch ein Gerät für knapp über 10 € kann auch mal verloren gehen, der finanzielle Verlust ist zu verschmerzen. Damit werden Anwendungen möglich, bei denen mit dem Verlust des Gerätes zu rechnen ist.

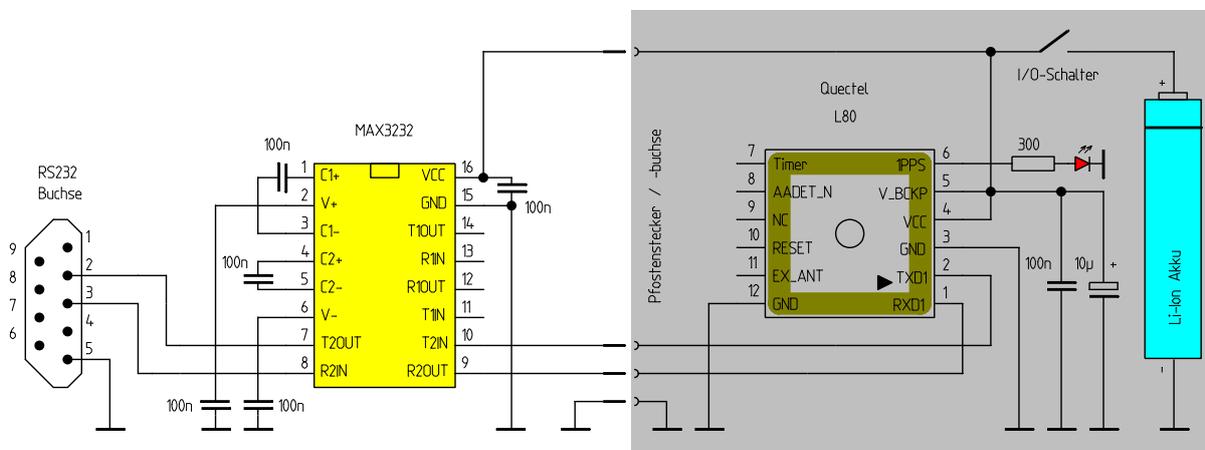


Bild 11 Der eigentliche GPS-Logger (rechts) kann kleiner als eine Streichholzschachtel werden

In meinem Mustergerät habe ich statt einer Box mit 3 NiMH AAA Akkus einen 1000 mAh Li-Ion-Akku vorgesehen. Da der Logger im Mittel nur 25 mA beansprucht, reicht für die 16 h Speicherkapazität auch ein kleinerer 500 mAh Li-Ion-Akku aus. Damit lässt sich gegenüber dem verwendeten Akku nochmal Platz und Gewicht sparen.

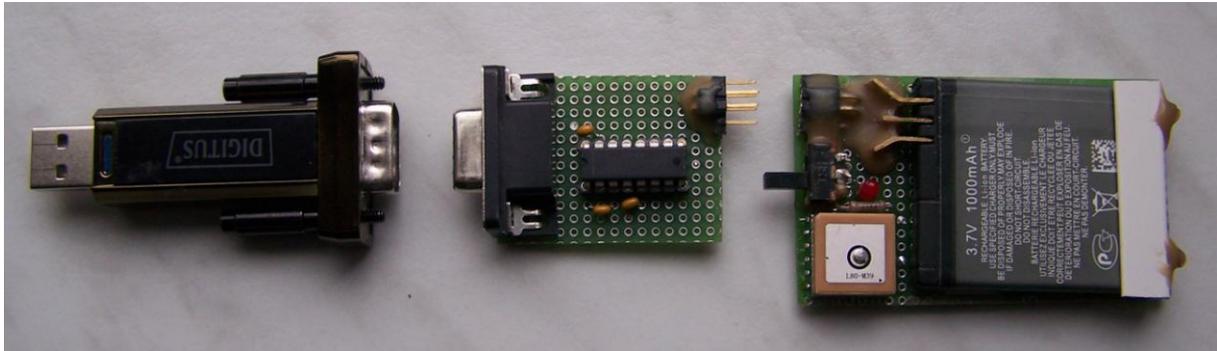


Bild 12 Getrennt – USB/Seriell-Wandler, RS232/TTL-Wandler, minimaler GPS-Logger

Den linken Teil der Schaltung, der Wandler von TTL auf RS232 kann man auch fertig kaufen, für „sagenhafte“ 0,05 € bei:

https://www.amazon.de/MAX3232-Serial-Konverter-Module-Stecker/dp/B00HV4K9ZY/ref=sr_1_49?m=AL7OCEVQ2O38D&s=merchant-items&ie=UTF8&qid=1487686806&sr=1-49

Hinweis:

Allerdings hat die chinesische Firma TS Trade, die hier bei Amazon anbietet, ein etwas merkwürdiges Geschäftsmodell, auf das hier hingewiesen werden soll. Der Preis von 0,05 € ist wirklich spektakulär, die Versandkosten von 3,00 € sind auch noch im grünen Bereich. Was man aber beachten sollte, die „Versandkosten“ verstehen sich pro Stück! Man wird bei solchen Pfennigartikeln schnell verleitet, gleich mal 10 Stück zu kaufen und hat dann bei lediglich 0,50 € Warenwert Versandkosten von 30,00 €. Also aufgepasst!



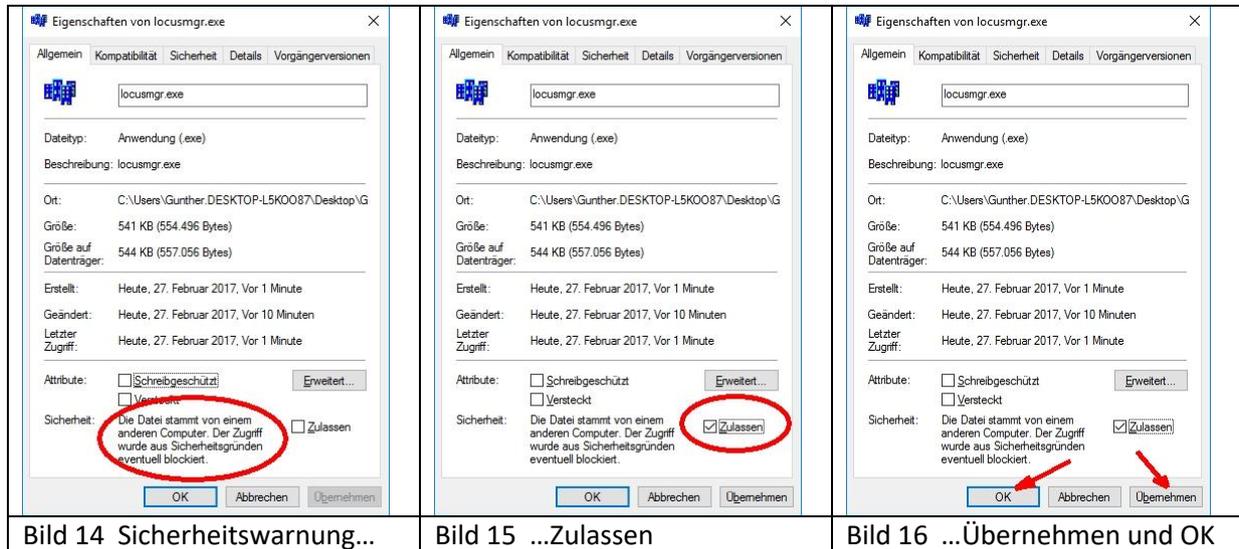
Bild 13 kommerzielle Variante des Pegelwandlers TTL/RS232

Eine weitere von Ernst B* im mikrocontroller.net vorgeschlagene Lösung beinhaltet den direkten Übergang von USB auf Serial(TTL) mit nur einem Modul. Damit wird ggf. Strom gespart, der DIGITUS-Konverter und der TTL/RS232-Wandler können entfallen:

<https://www.amazon.de/Aihasd-Converter-Module-Built-CP2102/dp/B01I4ZY5BW>

Probleme mit dem Programm locummgr.exe

Besonders unter Windows8 und Windows10 kann es vorkommen, dass das Programm locummgr.exe nicht gestartet werden kann. Dies liegt an einer Sicherheitseinstellung von Windows. Das Problem lässt sich beheben, wenn man die gespeicherte EXE mit der rechten Maustaste anklickt und Eigenschaften auswählt. Dann das Häkchen nach der Sicherheitswarnung aktivieren, auf Übernehmen und dann OK klicken:



Eine praktische Ergänzung

Es ist oft zweckmäßig, den Logger „vor Ort“ starten zu können und nicht auf einen PC für die notwendigen Start- und Stopp-Kommandos angewiesen zu sein. Warum nicht doch einen kleinen Mikrocontroller für diesen Zweck einsetzen? Das Problem bei so einer Steuerung ist, dass der L80 fortwährend seine NMEA-Signale aussendet und damit die serielle Kommunikation behindert. Nicht aber so bei der o.a. getrennten Ausführung vom eigentlichen L80-Logger und der USB/RS232/TTL-Seriell-Wandler-Schaltung. Hier wird der Logger ja ohne den hierfür hinderlichen MAX3232 betrieben.

Soll der Logger also mittels µC gestartet und gestoppt werden, muss dieser anstelle des PC's die seriellen Kommandos senden. Dazu wird lediglich anstelle der Wandler-Schaltung ein kleines Platinchen mit einem ATtiny13 und 2 Tasten (Start Log und Stopp Log) angesteckt. Bei angeschalteter Stromversorgung liefert der Akku des Loggers gleich Strom für den µC. Der macht zunächst gar nichts, sondern wartet auf einen Tastendruck. Je nach Taste sendet er nun an PB0 die für den L80 die benötigten Kommandos:

Taste „L80_start“ an PB1 → "\$PMTK185,0*23<CR><LF>"
 Taste „L80_stop“ an PB2 → "\$PMTK185,1*23<CR><LF>"

Die Schaltung für die Kommando-Platine ist sehr einfach, sie besteht nur aus dem ATtiny13, 2 Tasten, Pfostenstecker für GND, Pluspol Akku, TX und RX. Ein Quarz ist nicht notwendig, für das kurze Kommando reicht der (nicht sehr genaue) interne Oszillator auch bei 9600Bd aus, obwohl eigentlich nur Baudraten von bis zu 4800Bd empfohlen werden.

Die kleine Start/Stop-Platine könnte z.B. fest am Fahrrad montiert werden, während der Logger normalerweise zuhause ist. Vor Fahrtbeginn wird der Logger-Teil dann eingesteckt und angeschaltet. Dadurch bekommt nun auch der µC Betriebsspannung. Jetzt kann der Startknopf betätigt werden und der Logger startet. Während der Fahrt kann der Logger wieder abgezogen werden, um Strom zu

sparen (der μC braucht auch etwas...). Es stört aber die Funktion auch nicht, wenn Logger und μC verbunden bleiben. Am Ende der Fahrt wird dann die Stopp-Taste gedrückt und danach der Logger-Teil abgezogen - Fertig.

Die Auswertung allerdings braucht wieder einen PC – aber das ist bei praktisch allen Loggern so und damit kein Nachteil.

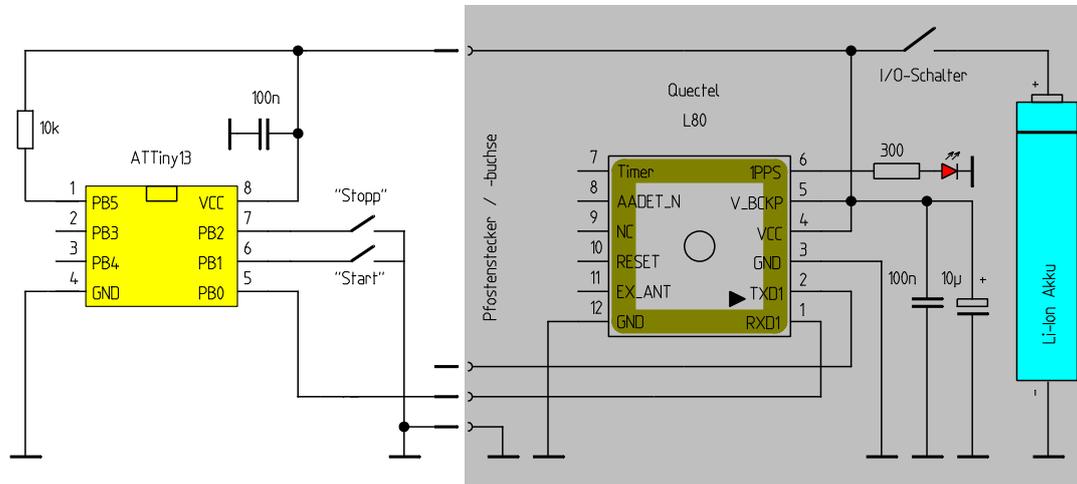


Bild 17 Start/Stopp-Schaltung ohne PC

Das BASCOM-AVR-Programm für den ATTiny13 ist simpel:

```

' BASCOM-Programm "L80_03.bas"
' Ansteuerung des L80 GPS-Loggers (Start- und Stopp-Kommando"

$regfile = "attiny13.dat"
$crystal = 1200000
$hwstack = 10
$swstack = 10

Config Pinb.1 = Input
L80_start Alias Pinb.1
Portb.1 = 1

Config Pinb.2 = Input
L80_stopp Alias Pinb.2
Portb.2 = 1

Wait 1

Open "COMB.0:9600,8,N,1" For Output As #1

Do
  If L80_start = 0 Then
    Waitms 100
    Print #1 , "$PMTK185,0*22<CR><LF>";

    Waitms 100
  End If

  If L80_stopp = 0 Then
    Waitms 100
    Print #1 , "$PMTK185,1*23<CR><LF>";

    Waitms 100
  End If
Loop

End

```

'ATTiny13 Konfiguration
'Taktfrequenz 1,2 MHz
'Stack einstellen
'Stack einstellen
'PinB.1 ist Eingang
'Start-Taste
'Pullup einstellen
'PinB.2 ist Eingang
'Stopp-Taste
'Pullup einstellen
'1 Sekunde warten
'Serielle Schnittstelle auf PinB.0
'9600Bd, 8Bit, No Parity,1 Stopbit
'als Output #1 deklarieren
'Do/Loop Schleife
'wenn Starttaste gedrückt
'100 ms warten
'Start-Kommando für L80 seriell
'ausgeben
'100 ms warten
'Ende Abfrage Starttaste
'wenn Stopptaste gedrückt
'100 ms warten
'Stopp-Kommando für L80 seriell
'ausgeben
'100 ms warten
'Ende Abfrage Stopptaste
'Ende Loop Schleife
'Programmende

Fazit

Der Aufwand zum Bau und Betrieb des vorgestellten Loggers ist minimal, je nach Erfahrung mit Mikrocontrollern gibt es verschiedene Anwendungen und Betriebsweisen.

Viel Spaß beim Basteln und Loggen...