



MechLab Engineering - Wir schaffen Fakten.



Low Budget Car PC

Quick Start Guide

Paul Balzer, Dipl.-Ing.(FH)

MechLab Engineering UG (haftungsbeschränkt) – Marienstr. 20, 01067 Dresden

Tel.: +49 179/1285141 **Mail:** balzer@mechlab-engineering.de



Zusammenfassung

Der Low Budget Car PC ist ein kompaktes Gerät, welches das einfach Loggen physikalischer Sensordaten ermöglicht. Ausgestattet mit Beschleunigungs-, Drehraten- und GPS Sensorik sowie einem CAN-Modul, ist er universell einsetzbar. Insbesondere die individuellen Applikationsmöglichkeiten für eigene Anwendungen machen ihn zu einem flexiblen Partner.

Der Anschluss erfolgt über den Standard Zigarettanzünder (12-24V), die Datenaufzeichnung beginnt ohne weiteres Zutun, signalisiert durch schnelles Blinken.

INHALTSVERZEICHNIS

Zusammenfassung	2
Quick Start	3
Sensoren	3
Erfasste Werte	3
CAN Daten	5
Log Dateien	5
Automatisierte Weiterverarbeitung	5
Zusätzliche Algorithmen	5
Eigene Entwicklung	6

Quick Start

1. Spannungsversorgung herstellen durch Bordnetz (Zigarettenanzünder, 6V...27V möglich)
2. ca. 20s warten auf Initialisierung
3. schnelles, dauerhaftes Blinken (durch Deckel) signalisiert loggen
4. Beenden durch Deaktivieren der Spannungsversorgung
5. beim nächsten Starten werden .zip Logdateien auf SD Karte zur Verfügung gestellt

Sensoren

Es sind Sensoren der Firma Tinkerforge GmbH verbaut. Die entsprechenden Datenblätter sind auf der Webseite von Tinkerforge¹ verfügbar.

IMU



- Voll ausgestattetes AHRS mit 9 Freiheitsgraden
- Keine akkumulierenden Fehler, kein Gimbal Lock!
- Vorkalibriert, einfach Anwendungsspezifisch zu kalibrieren
- Berechnet Quaternionen sowie Roll-, Nick- (Pitch) und Gier- (Yaw) Winkel

GPS



- Empfängt Bewegungs-, Positions-, Höhen- und Zeitdaten
- Interne Antenne, externe Antenne optional
- 66 Kanäle, 10Hz Update-Rate
- Hohe Empfindlichkeit und Genauigkeit, Störunterdrückung

Erfasste Werte

In der Standardkonfiguration loggt der Low Budget Car PC mit **50Hz**, je nach Ausstattung, folgende physikalischen Werte:

¹ <http://www.tinkerforge.com/>



Tabelle 1: Erfasste physikalische Werte

Wert	Einheit	Erläuterung	Name in CSV
Datum	-	Datum als ddmmyy	date
Zeit	-	Zeit als hhmmss	time
Timestamp	ms	Millisekunden seit 1.1.1970	millis
Längsbeschleunigung	m/s ²		ax
Querb beschleunigung	m/s ²		ay
Vertikalbeschleunigung	m/s ²		az
Rollrate	°/s		rollrate
ierrate	°/s		yawrate
Nickrate	°/s		pitchrate
Rollwinkel	°		roll
Nickwinkel	°		pitch
Gierwinkel	°	nicht identisch mit Fahrtrichtung	yaw
Geschwindigkeit	km/h		speed
Fahrtrichtung	°		course
Längengrad	°	in Dezimalgrad (DD.ddddd)	latitude
Breitengrad	°	in Dezimalgrad (DD.ddddd)	longitude
Höhe	m	Über Normalnull	altitude
Positions genauigkeit	m		pdop
horizontale Präzision	m		hdop
vertikale Präzision	m		vdop
Positionsfehler	m		epe
Satelliten in Sicht	-		satellites_view
Satelliten genutzt	-		satellites_used
Temperatur	°C		temp

CAN Daten

Das Loggen von CAN-Daten erfolgt über die TINY-CAN-API der Firma MHS Elektronik.

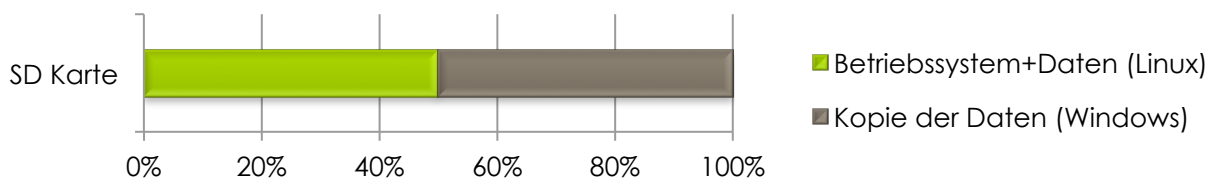
Beispiele, Dokumentation und Implementierungen findet man auf der Webseite der Firma MHS Elektronik².

Log Dateien

Die SD Karte ist standardmäßig ext4 formatiert, da die Funktion über ein Linux Betriebssystem ermöglicht wird.

Um eine einfache Handhabung für Windows-User zu ermöglichen, ist auf der SD Karte zusätzlich eine FAT32 Partition angelegt, auf welcher die Log-Dateien ZIP komprimiert abgelegt werden. Dies geschieht beim nächsten Start automatisch, kann aber eine gewisse Zeit in Anspruch nehmen, je nachdem wie groß die Log-Datei ist.

Es verringert sich der verfügbare Speicherplatz auf ca. die $\frac{1}{2}$ der angegebenen SD-Karten Größe.



Automatisierte Weiterverarbeitung

Die Log Dateien können anschließend automatisiert weiterverarbeitet werden. Denkbar (aber standardmäßig nicht konfiguriert) sind folgende Szenarien denkbar:

- Upload auf FTP Server
- Upload an Dropbox Account
- Upload an MechLab Engineering DataViewServer³
- Automatisches Generieren von Diagrammen

Zusätzliche Algorithmen

Die Daten können beim Loggen automatisch fusioniert werden. Dazu stehen optional Algorithmen zur Verfügung, welche die Positionsgenauigkeit verbessern⁴.

² <http://www.mhs-elektronik.de/index.php?module=download&action=list>

³ <http://mechlab-engineering.de/references/dataviewserver/>

⁴ <http://mechlab-engineering.de/2014/03/verbesserte-positionsschaetzung-durch-kalman-filter/>



Eigene Entwicklung

Auf dem CarPC können auch eigene Algorithmen appliziert werden. Dazu ist es notwendig sich in den CarPC einzuloggen.

Die einfachste Variante ist folgende:

Deckel öffnen, RaspberryPi leicht heraus heben, Bildschirm via HDMI anschließen, USB Tastatur einstecken. Nach dem Booten Login mit

Username: pi
Kennwort: raspberry

Alternativ kann der CarPC auch im eigenen WLAN eingerichtet werden:

Den Deckel des CarPC öffnen, ein WLAN USB Dongle (Empfehlung: EDIMAX EW-7811UN Wireless USB Adapter) in einen freien USB Port des RaspberryPi stecken. Anschließend über das eigene WLAN mit dem CarPC verbinden.

Die SD Karte mit einem SD Kartenleser (Windows, Mac, Linux) lesen und die Datei

`/boot/wlan.conf`

öffnen. Diese enthält die Login-Konfiguration für das WLAN. Typisch z.B. folgende Konfiguration (Router mit DHCP, also automatischer IP Vergabe, Netzwerk WPA2 Verschlüsselt):

```
network={  
  ssid="HIER_KOMMT_DER_WLAN_NAME_REIN_IN_ANFÜHRUNGSZEICHEN"  
  proto=RSN  
  key_mgmt=WPA-PSK  
  pairwise=CCMP TKIP  
  group=CCMP TKIP  
  psk="HIER_KOMMT_DAS_WLAN_KENNWORT_REIN_IN_ANFÜHRUNGSZEICHEN"  
}
```

Nachdem die Datei gespeichert wurde, den CarPC vom Strom trennen und neu verbinden. Nach einer kurzen Wartezeit sollte der Rechner „CarPC“ im Netzwerk auftauchen und man kann via SSH auf den CarPC zugreifen:

```
ssh pi@CarPC
```

Username: pi
Kennwort: raspberry
