

Auswertung der Daten des HAC4

Die Fahrradcomputer vom Typ HAC4 von der Firma CicloSport zeichnen bei Bedarf Daten (Höhe, Weg, Herzfrequenz, Temperatur und Trittfrequenz) auf. Diese Daten können mit einem Interface über eine serielle Schnittstelle zu einem PC übertragen werden. Die nachfolgende Dokumentation beschreibt das Auslesen des HAC4 und die Formatierung der ausgelesenen Daten. Es ist zu beachten, dass es drei Versionen des HAC4 gibt, den HAC4-325, HAC4-315 und den HAC4-Imp. Die Unterschiede im Format der Daten sind dokumentiert.

Serielle Schnittstelle

Die serielle Schnittstelle des HAC4 ist unidirektional ausgeführt, das heißt, es können nur die Daten in einer Richtung von der Uhr zum PC übertragen werden. Es ist kein spezielles Protokoll implementiert. Zum Kontrollieren der Datenübertragung muß die abschließend gesendete Checksumme herangezogen werden.

Die einzustellenden Parameter der seriellen Schnittstelle sind:

Geschwindigkeit:	9600 bps
Datenbits:	8
Stopbit:	1
Parität:	keine

Für die Übertragung an sich wird keinerlei Flußkontrolle verwendet. Es ist allerdings notwendig, die serielle Schnittstelle auf Flußkontrolle ‚Hardware‘ einzustellen, damit das Interface die Daten ordnungsgemäß zum PC weiterleiten kann.

Datenübertragung

Die Datenübertragung wird am HAC4 gestartet. Als erstes wird die Startkennung, danach werden die eigentlichen Nutzdaten und am Ende wird die Checksumme übertragen.

Die Startkennung besteht aus den vier Byte ‚AFRO‘, die durch ein <CR> abgeschlossen werden (0x41, 0x46, 0x52, 0x4f, 0x0d). Eine Software die den HAC4 ausliest, sollte den Start der Übertragung auf die Zeichenkette ‚FRO<CR>‘ synchronisieren. Es ist möglich, daß vor der Kennung ‚AFRO‘ unbestimmte Zeichen gesendet werden. Auch kann es sein, dass das einführende ‚A‘ von ‚AFRO<CR>‘ verändert gesendet wird.

Nach der Startkennung werden die Nutzdaten übertragen. Die Nutzdaten repräsentieren den internen Speicher des HAC4 mit einer Größe von 32KByte und enthalten unter anderem die aufgezeichneten Daten.

Jeweils zwei Byte werden hexadezimal in vier Byte umgewandelt. Diese werden dann übertragen und durch ein <CR> abgeschlossen. Das heißt für insgesamt zwei Bytes werden 5Byte gesendet. Also für die zwei Byte 0xb7 und 0x35 würde zum Beispiel die Zeichenkette ‚B735<CR>‘ gesendet werden.

Insgesamt werden Nutzdaten von 16.384 mal 5 Zeichen übertragen.

Als Abschluß wird eine zwei Byte Checksumme im gleichen Format (5Byte) gesendet.

Die Checksumme wird berechnet, indem die gesamten Nutzdaten als vorzeichenbehaftete 16 Bit Integer Werte addiert werden.

Datenformat

Die Aufteilung der Nutzdaten (32KByte) erfolgt in Einheiten zu 16 Bit. Daraus ergibt sich ein adressierbarer Bereich von 0x0000 bis 0x3fff.

Die für die Auswertung wichtigen Daten stehen ab Adresse 0x0080. An dieser Adresse befindet sich bei den Versionen HAC4-315 und HAC4-Imp der Parameterspeicher. Er ist wie folgt formatiert:

```

0x0080h -----
  unsigned short magic;           // 0xb735 für HAC4-315, 0xb734 für HAC4-Imp
  unsigned short umfang;         // Umfang des Rades
  unsigned short gewicht;        // Gewicht
  short homeAlti;                // Home Höhe
  short otg1;                    // erste Obergrenze für Herzfrequenz
  short utg1;                    // erste Untergrenze für Herzfrequenz
  short otg2;                    // zweite Obergrenze für Herzfrequenz
  short utg2;                    // zweite Untergrenze für Herzfrequenz

0x0088h -----
  unsigned short count1;         // Count Down Zähler 1
  unsigned short count2;         // Count Down Zähler 2
  short korrAlti;                // Korrekturwert der Höhe
  short totDistH;                // gefahrene Kilometer HIGHWORD
  short totDistL;                // gefahrene Kilometer LOWWORD
  short ptrStart;                // Startzeiger der Tourenaufzeichnung
  short year;                    // eingestelltes Jahr im BCD Format
  char mon;                      // eingestellter Monat im BCD Format
  char day;                      // eingestellter Tag im BCD Format

0x0090h -----
  unsigned short meterAufwaerts; // Aufgelaufene Höhenmeter aufwärts
  unsigned short meterAbwaerts; // Aufgelaufene Höhenmeter abwärts
  short maxAlti;                 // maximale Höhe Tageswert
  short hour;                    // Jahresfahrzeit Stunden im BCD Format
  char min;                      // Jahresfahrzeit Minuten im BCD Format
  char sec;                      // Jahresfahrzeit Sekunden im BCD Format
  short ptrLog;                  // interner Zeiger
  short ptrEnd;                  // interner Zeiger
  short ptrStop;                 // interner Zeiger

```

Die Parameter enthalten die Grundeinstellungen des HAC4. Für die Auswertung hat in erster Linie der Parameter `magic` an der Adresse 0x0080 Bedeutung. Mit Hilfe des `magic` kann auf das vorliegende Gerät geschlossen werden.

Folgende Magic-Codes sind bis heute implementiert:

0xb735:	HAC4-315
0xb7b4:	HAC4-Imp
sonst:	HAC4-325

Zu beachten ist, dass die HAC4-325 noch keinen Parameterblock besessen haben. Kann dem `magic` weder der HAC4-315 noch der HAC4-Imp zugeordnet werden, so ist davon auszugehen, dass es sich um Daten eines HAC4-325 handelt. Die richtige Zuordnung ist vor Beginn der Auswertung der Aufzeichnung wichtig, da sich das Format der einzelnen Versionen geringfügig unterscheidet.

Bei den Versionen HAC4-315 und HAC4-Imp schließt sich nach dem Parameterspeicher ab Adresse 0x0098 der Speicher für die Aufzeichnungen an. Beim HAC4-325 beginnt der Speicher bereits ab Adresse 0x0090. Der Speicher für die Aufzeichnungsdaten (von 0x0098 bzw. 0x0090 bis 0x3fff) ist als Ringspeicher organisiert. In diesem Ringspeicher werden die Aufzeichnungsdaten von Touren in Blöcken zu 16 Byte nacheinander abgespeichert.

Aufzeichnung einer Tour

Die Aufzeichnung einer Tour wird durch den Startblock eingeleitet. Danach schließen sich je nach Aufzeichnungsdauer eine Anzahl von Logblöcken an. Abschließend gehört zu jeder Aufzeichnung noch ein Endblock sowie ein Stopblock. Jeder Block hat eine Größe von 16 Byte und wird durch einen Id eindeutig gekennzeichnet. Damit kann in dem Aufzeichnungsspeicher gezielt nach einem einleitenden Startblock gesucht werden. Jeder Logblock enthält die Daten der

vergangenen 2 Minuten. Der Endblock enthält die Daten, welche nach dem Speichern des letzten Logblockes angefallen sind (maximal 1 Minute und 59 Sekunden). An Hand der Anzahl der Logblöcke zwischen dem Startblock und dem Endblock und dem Feld `time` aus dem Endblock kann die Dauer der Aufzeichnung festgestellt werden.

Startblock

Das Format eines Startblockes wird durch die unten stehende Struktur beschrieben:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
id = 0xA				res0				res1				mode		res2	
endptr															
min								hour							
day								mon							
reserve															
reserve															
alti															
pulse															

```
typedef struct tag_HAC4StartBlock{
    unsigned short id:4;          // Identifier 0xA für Startblock
    unsigned short res0:4;       // Reserve
    unsigned short res1:4;       // Reserve-Bits
    unsigned short mode:2;       // Modus Joggen, Ski, Bike
    unsigned short res2:2;       // Reserve-Bits
    unsigned short endptr;       // Byte-Zeiger auf STOPBLOCK der Aufzeichnung
    unsigned char min;           // Aufzeichnung: Uhrzeit Minuten
    unsigned char hour;          // Aufzeichnung: Uhrzeit Stunden
    unsigned char day;           // Aufzeichnung: Datum Tag
    unsigned char mon;           // Aufzeichnung: Datum Monat
    unsigned int reserve;        // Reserve
    short alti;                  // Ausgangshöhe
    short pulse;                 // Ausgangspuls
}THAC4StartBlock;
```

- id:** Der Id ist bei einem Startblock auf den Wert 0xA gesetzt. Jeder Startblock kann eindeutig durch den Id identifiziert werden.
- mode:** Mode zeigt an, im welchen Modus der HAC4 während der Aufzeichnung stand.
- 0: Joggen
 - 1: Ski
 - 2: Bike
 - 3: Bike - Ski
- endptr:** zeigt auf den abschließenden StopBlock. Der Stopblock wiederum hat ein Datenfeld, welches auf den Startblock verweist. Eine Tour ist nur dann korrekt aufgezeichnet, wenn der Startblock auf den Endblock und der Endblock auf den Startblock zeigt.
- min:** Beginn der Aufzeichnung Minute der Uhrzeit (BCD-Format)
- hour:** Beginn der Aufzeichnung Stunde der Uhrzeit (BCD-Format)
- day:** Beginn der Aufzeichnung Tag des Monats (BCD-Format)
- mon:** Beginn der Aufzeichnung Monat (BCD-Format)
- alti:** Höhe bei Aufzeichnungsbeginn. Bei der weiteren Aufzeichnung wird nur die Änderung der Höhe abgespeichert.
- pulse:** Herzfrequenz bei Aufzeichnungsbeginn. Bei der weiteren Aufzeichnung wird nur die Änderung der Herzfrequenz abgespeichert.

Logblock

Das Format eines Logblockes wird in den unten stehenden Datenstrukturen beschrieben:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
id = 0xB				res0				temp							
cad								lap							
dist20				alti20				pulse20							
dist40				alti40				pulse40							
dist60				alti60				pulse60							
dist80				alti80				pulse80							
dist100				alti100				pulse100							
dist120				alti120				pulse120							

```
typedef struct tag_HAC4LogValue{
    unsigned short dist:6;          // Änderung des Weges in Vielfachen von 10 Metern (0..630 m)
    short alti:6;                  // Wert für die Änderung der Höhe (-128..121 m)
    short pulse:4;                 // Änderung der Herzfrequenz in 1/min (-8..+7 1/min)
}THAC4LogValue;

typedef struct tag_HAC4LogBlock{
    unsigned char id:4;             // Identifier B
    unsigned char res0:4;          // Reserve
    char temp;                     // Temperatur
    unsigned char cad;             // Trittfrequenz
    unsigned char lap;             // Zeit an der Marke gesetzt wurde
    THAC4LogValue val[6];         // 6 Datensätze (dist, alti, pulse) für jeweils 20 Sekunden
}THAC4LogBlock;
```

Der Logblock enthält die Daten von 2 Minuten. Das sind die Trittfrequenz und die Temperatur als jeweils ein Wert für die zwei Minuten und die Weg-, die Höhen- und die Herzfrequenzänderung für jeweils 20 Sekunden. Außerdem kann während der 2 Minuten eine Marke gespeichert sein.

- id: Der Logblock hat immer den Id 0xb
- temp: Temperatur für 2 Minuten
- cad: Trittfrequenz für 2 Minuten. Der HAC4-325 kann keine Trittfrequenz aufzeichnen. Bei dieser Version ist der Inhalt des Feldes cad unbestimmt.
- lap: Zeit, an der eine Marke gesetzt wurde. Beim HAC4-325 ist die Zeit als ein Vielfaches von 10 Sekunden (0..12) zu verstehen. Bei den anderen Versionen ist die Zeit sekundengenau gespeichert (0..120 Sekunden). Ist der Wert auf Null gesetzt, ist keine Marke vorhanden.
- dist: Änderung des Weges während der letzten 20 Sekunden in Vielfachen zu 10m. Da für diesen Parameter 6Bit Speicher vorgesehen sind, können maximal 630m, aufgezeichnet werden.
- alti: Wert für die Änderung der Höhe während der letzten 20 Sekunden. Die Höhenänderung in Metern dh wird durch folgenden Zusammenhang gebildet.
- | | | |
|-----|----------------|-----------------------|
| -15 | <= alti <= 15: | dh = alti |
| -32 | <= alti < -15: | dh = (alti+16)*7 - 16 |
| 15 | < alti <= 31: | dh = (alti-16)*7 + 16 |
- Aus diesem Zusammenhang ergibt sich eine maximale Höhenänderung von -128m bis 121m.
- pulse: Änderung der Herzfrequenz während der letzten 20 Sekunden in 1/min.

Endblock

Das Format eines Endblockes wird in der unten stehenden Datenstruktur beschrieben:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
id = 0xC				res0				temp							
res1								time							
dist20				alti20				pulse20							
dist40				alti40				pulse40							
dist60				alti60				pulse60							
dist80				alti80				pulse80							
dist100				alti100				pulse100							
dist120				alti120				pulse120							

```
typedef struct tag_HAC4EndBlock{
    unsigned char id:4;           // Identifier C
    unsigned char res0:4;        // Reserve
    char temp;                   // Temperatur
    unsigned char res1;          // Reserve (HAC4-325: entspricht time)
    unsigned char time;          // Zeit vom letzten Logblock in Sekunden (0..119)
    THAC4LogValue val[6];        // maximal 6 Datensätze (dist, alti, pulse)
}THAC4EndBlock;
```

id: Der Endblock hat immer den Id 0xc

temp: Temperatur für 2 Minuten

time: Zeit in Sekunden seit der Speicherung des letzten Logblockes. Aus diesem Feld ergibt sich, wie viele Datensätze mit den Weg, Höhen- und Herzfrequenzänderungen ausgewertet werden müssen. Der letzte auszuwertende Datensatz bezieht sich auf eine Zeit von time modulo 20 Sekunden.

dist: Änderung des Weges während der letzten 20 Sekunden (bzw. time modulo 20) in Vielfachen zu 10m. Da für diesen Parameter 6Bit Speicher vorgesehen sind, können maximal 630m, aufgezeichnet werden.

alti: Wert für die Änderung der Höhe während der letzten 20 Sekunden (bzw. time modulo 20). Die Höhenänderung in Metern wird durch folgenden Zusammenhang gebildet.

```
-15 <= alti <= 15: dh = alti
-32 <= alti < -15: dh = (alti+16)*7 - 16
15 < alti <= 31: dh = (alti-16)*7 + 16
```

Aus diesem Zusammenhang ergibt sich eine maximale Höhenänderung von -128m bis 121m.

pulse: Änderung der Herzfrequenz während der letzten 20 Sekunden (bzw. time modulo 20) in 1/min.

Stopblock

Das Format eines Stopblockes in der unten stehenden Datenstruktur beschrieben:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
id = 0xD				res0				res1							
startptr															
reserve															
reserve															
reserve															
reserve															
reserve															
reserve															

```
typedef struct tag_HAC4StopBlock{
    unsigned char id:4;           // Identifier D
    unsigned char res0:4;        // Reserve
    unsigned char res1;          // Reserve
    unsigned short startptr;     // Byte-Zeiger auf STARTBLOCK der Aufzeichnung
    unsigned short res2[6];      // Reserve
}THAC4StopBlock;
```

id: Der Id ist bei einem Stopblock auf den Wert 0xd gesetzt.

startptr: Zeigt auf den Startblock. Eine Tour ist nur dann korrekt aufgezeichnet, wenn der Startblock auf den Endblock und der Endblock auf den Startblock zeigt.