

Preface

In September 2009 the prices for the Suunto M9 GPS wristop computer had dropped to a level I could not resist to get one. If you know that it is already available for a few years the features are still impressive. The biggest downside apart from the slow GPS acquisition (for today's standards) is the missing software for people without MS Windows. This is also true for the Suunto models X9, X9i, X10 and G9.

So I decided to write a proof of concept in Java that is portable to platforms like Sun Solaris 10, Mac OS X (ppc and x86) and Linux. The proof of concept actually supports Windows as well as all of the above except Linux (not tried yet).

Before I could write the proof of concept I had to reverse engineer the protocol used by the Suunto GPSes. While doing this I discovered a lot of similarities between the M9 and the X10 (X9, X9i being identical to the X10) but also a lot of differences.

The following article describes the Suunto protocols for the M9 and X10 GPS wristop computers to the best of my knowledge. It is not complete and might be in many cases be incorrect. But it will be a good starting point for others.

If you find any mistakes, typos or if you have further informations on the G9, X9 or X9i let me know.

March 2011

Gerold Pauler (suunto@paulerplan.de)

Disclaimer

The information in this text is to the best of my knowledge. It is not complete and may be misleading or even incorrect. I will give no warranty for correctness or usability in any form. I also will not take any responsibility for damages that might occur to your Suunto wristop computer while trying to use the informations herein.

Credits

gpsd project <http://gpsd.berlios.de/index.html>

Markus Fritze http://www.sarnau.info/papers:suunto_protocol

(iTalk protocol Rev 1.5)

(Suunto dive computer protocols)

Document History

Version 1	2011-03-27	first published Version
Version 2	2011-04-02	Credits and minor corrections
Version 0.3	2011-04-20	Major number changed to 0 to show document is still work in progress. History added and many corrections on routes and way points

Inhaltsverzeichnis

Preface.....	1
Disclaimer.....	1
Credits.....	1
Document History.....	1
Suunto Wristop Computers Comparison.....	3
General features.....	3
Tracks, Routes and Way Points.....	4
Routes.....	4
Way points.....	4
Tracks.....	4
Track points.....	4
Suunto M9, (G9?), X9, X9i, X10 Protocol.....	5
General Message Format.....	5
iTalk Header.....	5
iSys Header.....	5
Suunto Message Types.....	6
iTalk Data Format.....	7
Get UKHO User Permit (M9 only).....	7
Get List of Way Points for Route ID.....	7
Get Info about Routes.....	8
Get Way Point (MyPoints, MOB, Route Points).....	8
Get Route (MyPoints, Routes and Race Courses).....	9
Delete Route.....	9
Delete Way Point.....	9
Disconnect Way Point from Route.....	9
Set Way Point.....	10
Connect Way Point to Route.....	10
Set Route (Routes and Race Courses).....	11
End of Route Setup.....	11
Delete All.....	11
Get Number of Tracks.....	12
Get Track.....	12
Delete All Tracks.....	12
Get Track Point.....	13
Get Firmware Revision (Part I).....	13
Use of dive computer format in iTalk data.....	14
Read memory - reads memory from the Suunto.....	14
Write memory - changes the memory inside the Suunto.....	14
Get firmware revision (Part II).....	14
Get Memory Usage.....	15
Memory Map.....	16
Examples of Memory Locations.....	28
Examples of Annotated Protocol Tracelogs.....	29

Suunto Wristop Computers Comparison

General features

Model	M9	G9	X9	X9i	X10
Activity	Sailing	Golf	Outdoor	Outdoor	Outdoor
Interface (PC)	serial	serial	serial	USB	USB
Interface (WC)	serial	serial	serial	serial	serial
GPS Performance	poor	poor	poor	acceptable	good
Identifier	0x5b	???	0x5e?	0x5e?	0x5e
Protocol	M9	G9	X10	X10	X10
Number of Routes	50	n/a	50	50	50
Number of Way Points	500	n/a	500	500	500
Number of Tracks	25	n/a	25	25	25
Number of Track Points	8000	n/a	8000	8000	26000

Although I only have access to a M9 and a X10 from the traced messages of the Suunto Trek Manager software I am sure that the protocol for the X9, X9i and X10 must be identical.

Tracks, Routes and Way Points

The following information about tracks, routes and way points are from the different manuals and technical descriptions.

It is obvious, that the informations therein are NOT consistent.

Routes

Maximum 50 routes (3 or 10 race routes M9)

Special routes:

Route Id 1: MyPoints

Route Id 48: Race 1 (M9)

Route Id 49: Race 2 (M9)

Route Id 50: Race 3 (M9)

Way points

Maximum 500 way points

Maximum 50 way points per route

Special way points:

Way point Id 490: MOB (M9), Home (X9, X9i, X10)

Way point Id 491: Alarm point 1 (X9,X9i,X10)

...

Way point Id 500: Alarm point 10 (X9,X9i,X10)

Tracks

Maximum 25 tracks

Track points

Maximum 8000 track points (M9, X9, X9i)

26000 track points (X10)

Suunto M9, (G9?), X9, X9i, X10 Protocol

General Message Format

All messages look like:

```
3c 2a ll ll xx xx yy yy zz zz ... .. pp pp 3e    <*>
```

Meaning:

```
3c 2a "<*" Message start
ll ll   length of payload in 16 bit words
xx xx   llll words payload: consisting of 5 words iSys header,
yy yy   rest of 4 words iTalk header,
zz zz   llll - 9 words iTalk data
pp pp   Checksum - simple addition of iTalk data
3e ">" Message end
```

According to iTALK protocol specification Rev. 1.5 (only chapters 1 to 3 are relevant)

iTalk Header

- iSys Header (5 words)
- Source (1 word)
- Destination (1 word)
- Transaction ID (1 word)
- Data Length (1 word)

iSys Header

0x0000	Reserved	(Suunto sets it to arbitrary)
0x0000	Reserved	(Suunto sets it to arbitrary)
0x0000	Reserved	(Suunto sets it to arbitrary)
0x01xy	xy defines message type	(Suunto mostly uses 0x03xy - iTalk Protocol only specifies 0x01xy)
0x0000	Host -> iTrax	(arbitrary iTrax -> Host)

further explanation of iTalk header see iTalk protocol specification Rev. 1.5.

You can find the protocol specification at

<http://code.google.com/p/gpsd/downloads/detail?name=iTalk-1.5.pdf>

or <http://gpsd.googlecode.com/files/iTalk-1.5.pdf>

Suunto Message Types

0x012E	Get UKHO user permit (M9 only)
0x0321	Get list of way points for route id
0x0322	Get info about routes, ...
0x0323	Get way point (MyPoints, MOB, route points)
0x0324	Get route (MyPoints, routes and race courses)
0x0326	Delete route
0x0327	Delete way point
0x0328	Disconnect way point from route
0x0329	Set way point
0x032A	Connect way point to route
0x032B	Set route (routes and race courses)
0x032C	End of route setup
0x032D	Delete all
0x0341	Get number of tracks
0x0342	Get track
0x0346	Delete all tracks
0x0347	Get track point
0x03F1	Get firmware revision
0x03F3	Use of dive computer format in iTalk data
0x03F4	Get memory usage

--- Not traced yet – do these exist ? ---

0x034? Delete track

0x034? Delete track point

--- Not traced yet – do these exist ? ---

iTalk Data Format

Get UKHO User Permit (M9 only)

0x012E Get UKHO user permit (**M9 only**)
Where is the pin code stored? Or is it the same for all M9s?

39 (0x27) words
1 word: 0x0201
38 words: don't care

Answer is:
39 (0x27) words
1 word: 0x0200
4 words: 0x0000
2 words: 0xffff0, 0xffff meaning unknown
4 words: UKHO user permit
28 words: 0xffff

Get List of Way Points for Route ID

0x0321 Get list of way points for route id
53 (0x35) words
1 word: 0x0000
1 byte: 0x32 (50 way points)
1 byte: 0x00
1 word: Route ID
50 words: don't care

Answer is:
1 word: 0x0000
1 word: Number of way points (n)
1 word: Route ID
n words: Way point ID[s]

Way point ID 490: MOB (**M9**), Home (**X10**)
Way point ID 491: Alarm point 1 (**X10**)
...
Way point ID 500: Alarm point 10 (**X10**)

Attention: alarm points in X10 get an way point ID of (0x400 | way point ID) in “get list of way points for route id”. If you want to use it in “get way point” you have to strip this bit first (0x3ff & way point ID)

Get Info about Routes

0x0322 Get info about routes, ...
 102 (0x66) words
 1 word: 0x0000
 1 byte: 0x64 (100 = 50 routes * 2)
 1 byte: 0x00
 100 words: don't care

Answer is:
 1 word: 0x0000
 1 word: Number of routes (n)
 n words: Route ID[s]

Route ID 0x0001 is MyPoints
 Route IDs 0x0002 - 0x002F are user routes
 Route IDs 0x0030 - 0x0032 are race courses (race 1 - race 3 – **M9**)

Get Way Point (MyPoints, MOB, Route Points)

0x0323 Get way point (MyPoints, MOB, route points)
 28 (0x1c) words
 1 word: Way point ID
 27 words: don't care

Answer is:
 28 (0x1c) words
 1 word: Way point ID
 12 bytes: Way point name (bytes swapped)
 4 words: ???
 1 word: ??? (0x00ff)
 2 words: Latitude
 2 words: Longitude
 1 word: Altitude [m]
 1 word: Alarm Radius [m] (-1 if not alarm point)
 1 word: Pressure [hPa]
 5 words: don't care ?
 4 words: ???

Attention: alarm points in X10 get an way point ID of (0x400 | way point ID) in “get list of way points for route id”. If you want to use it in “get way point” you have to strip this bit first (0x3ff & way point ID)

Get Route (MyPoints, Routes and Race Courses)

0x0324 Get route (MyPoints, routes and race courses)
One word: Route ID

Answer is:
12 (0x0c) words (**M9**)
13 (0x0d) words (**X10**)
1 word: Route ID
20 bytes: Route name (bytes swapped)
1 word: Number of way points for route
1 word: ??? (**X10 only**)

Delete Route

0x0326 Delete route
One word
1 word: Route ID

Answer is one word
ACK (0x8000)

Delete Way Point

0x0327 Delete way point
One word
1 word: Way point ID

Answer is one word
ACK (0x8000)

Disconnect Way Point from Route

0x0328 Disconnect way point from route
2 words
1 word: Way point ID
1 word: Route ID

Answer is one word
ACK (0x8000)

Set Way Point

0x0329 Set way point
 28 (0x1c) words
 1 word: Way point ID (0x01ea = 490 MOB)
~~1 word: Type of way point (0x0000 for route points and race points)~~
 6 words: Name (max. 12 chars, race points end with "?")
 1 word: 0x0000 ???
 3 words: ???
 1 word: 0x00ff ???
 2 words: Latitude
 2 words: Longitude
 3 words: 0x0000 ???
 1 word: 0x0001 ???
 8 words: ???
 Answer is one word
 New way point ID

01 ea
 4f 4d 00 42 77 00 fa 08 00 12 00 18
 00 00
 fa a4 00 12 5d 60
 00 ff
 7b 1f 23 ed 1d 9d 0e e3
 00 00 00 00 00 00 00 01
 53 8b 72 f6 9d 1f 75 87 35 ef 75 87 28 d1 00 75

Connect Way Point to Route

0x032A Connect way point to route
 3 words
 1 word: Way point ID
 1 word: Route ID
 1 word: nth way point of route
 (n = -1 (0xffff) no order in route (MyPoints))

 Answer is one word
 ACK (0x8000)

After “set way point” and “connect way point to route” you have to wait for processing with the following procedure:

M9: Write 0x00 to Location 0x4000

Write 0x00 to Location 0x7920

X10: ???

Set Route (Routes and Race Courses)

0x032B Set route (routes and race courses)
 12 (0x0c) words
 1 word: Route ID (0x0001 = MyPoints, 0x0030 .. 0x0032 = race courses)
~~1 word: Type of route (0x0000 = normal route, 0x0030 = race course)~~
 n words: Name (chars of name - Attention bytes swapped)

Answer is one word
 New route ID

After “set route” you have to wait for processing by repeating the following procedure twice:

M9: Write 0x00 to Location 0x4000
Write 0x00 to Location 0x7920
X10: ???

End of Route Setup

0x032C End of route setup
 No word (length = 0x0000)

Answer is one word
 ACK (0x8000)

After “end of route setup” you have to wait for processing with the following procedure:

M9: Write 0x01 to Location 0x0000
X10: ???

Delete All

0x032D Delete all
 No word (length = 0x0000)

Answer is one word
 ACK (0x8000)

Get Number of Tracks

0x0341 Get number of tracks
 No word (length = 0x0000)

Answer is:
 1 ... n words
 1 word: Number of tracks
 2nd ... nth word: Track IDs (**X10** only)

Get Track

0x0342 Get track
 One word: Track ID

Answer is
 13 (0x0d) words
 1 word: Track ID
 10 words: Track name (Attention: bytes swapped)
 1 word: Number of track points
 1 word: ID of starting track point (**X10** always 1? maybe Number of
 Mempoins)

Delete All Tracks

0x0346 Delete all tracks
 No word (length = 0x0000)

Answer is one word
 ACK (0x8000)

After “delete all tracks” you have to wait for processing with the following procedure:

M9: Write 0x00 to Location 0x6000

Write 0x01 to Location 0x0093

X10: ???

Get Track Point

0x0347 Get track point
 1 word: ID of first track point (**M9** absolute / **X10** relative to track)
 1 word: number of track points (n)
 1 word: track ID (**X10** only)

Answer is

2 (0x02) + n * 15 (0x0f) words

1 word: ID of first track point

1 word: Number of track points

n * 15 words: data of the n track points

1 word: Figure Of Merit (FOM)

3 words: Date and time (UTC) (binary coded in 6 bytes)

1 word: Log-Id??? (always 255 ???)

2 words: Latitude (double word - low word, high word - divide by 10000000)

2 words: Longitude (double word - low word, high word - divide by 10000000)

1 word: Altitude [m]

1 word: Vertical speed [cm/s]??

1 word: Horizontal speed [cm/s]

1 word: True heading [1/10 degrees]

1 word: Sample time??? (always 0 ???)

1 word: Track point ID (**X10** always)

(**M9** if value == Track point ID then
 it is a Mepoint or Windpoint not a Trackpoint)

Get Firmware Revision (Part I)

0x03F1 Get firmware revision
 12 (0x0c) words
 1 word: 0x01
 1 word: 0x00
 10 words: don't care

Answer is:

12 (0x0c) words

1 word: 0x01

1 word: 0x05 relevant values

5 words: Firmware Rev. 2, 1, 11, 3281, 0 => 1.11.3281

5 words: don't care

Use of dive computer format in iTalk data

0x03F3 Use dive format in iTalk data (format is similar to Suunto dive computers like D9)
From Markus Fritze (MMM) http://www.sarnau.info/papers:suunto_d9 but slightly adjusted for M9 and X10

Algorithm for checksum calculation:

```
unsigned char checksum = 0x00;
for(int i=0; i<packageLen; ++i)
  checksum ^= package[i];
```

The package starts with a command byte, followed by the length of bytes for the parameters (0x00 0x03 = 3 Bytes), the actual parameters and the checksum (CS) at the end.

Here are examples of the most common commands:

Read memory - reads memory from the Suunto

- to Suunto: 0x05 + 0x00 + 0x03 + addr_high + addr_low + count (1..0x78) + CS
- from Suunto: 0x05 + 0x00 + (0x03 + count) + addr_high + addr_low + count (1..0x78) + count Bytes + CS

Write memory - changes the memory inside the Suunto

(WARNING: this can damage your watch!!!)

- to Suunto: 0x06 + 0x00 + (0x03 + count) + addr_high + addr_low + count (1..0x78) + count Bytes + CS
- from Suunto: 0x06 + 0x00 + 0x04 + addr_high + addr_low + count (1..0x78) + first Byte + CS

Get firmware revision (Part II)

- to Suunto: 0x0F + 0x00 + 0x00 + CS
- from Suunto: 0x0F + 0x00 + length (0x0A) + 0x76 + HH + HL + 0x2E + LH + LL + 0x00 + 0x17 + 0x00 + 0xFF + CS = "v"+HH+HL+"."+HL+LL
- Example: "v01.28"

Instead of 0x0F used by Trek Manager Sail Manager uses 0x07.

Read and Write Memory are used quite frequently by Sail-/Trek-Manager and M9, X9, X9i and X10 to read/change basic settings of the watch but also to read the barometer data, track header data and so on.

Suunto answers with an ACK package first.

It is an 0x03F3 answer with only one word 0x0080 for ACK.

Other answers are failures.

Then a second 0x03F3 answer with the actual response is send by the Suunto.

Get Memory Usage

0x03F4 Get Memory Usage
No word (length = 0x0000)

Answer is:
One word: Percentage of free memory

Attention:

To get all informations you often need a mix of protocol requests and reading of memory locations.

Memory Map

Legend: SM = Sail Manager, TM = Trek Manager

Address	Length	Description	Values	Default	0x00	0x01	0x02	0x03	0x04	0x05	0xff
0x0000	0x01	???		0x01							
0x0006	0x0d	Units Configuration (0x0e TM)									
0x0006	0x01	Clock 12h/24h		0x01	12h	24h					
0x0007	0x01	Date		0x01	mm.dd	dd.mm					
0x0008	0x01	Elevation (Altitude)		0x00	m	ft					
0x0009	0x01	Temperature		0x00	°C	°F					
0x000a	0x01	Pressure		0x00	hPa	inHg					
0x000b	0x01	Distance		0x02	km	mi	nm(SM)				(TM)
0x000c	0x01	Short Distance		0x03	n/a	???	???	m(SM)	ft	yd	
0x000d	0x01	Velocity		0x02	km/h	mph	kt(SM)	m/s(SM)			(TM)
0x000e	0x01	Wind Unit (SM)		0x03			kt	m/s			
0x000e	0x01	vertical Velocity (TM)		0x01	m/min	m/h	ft/min	ft/h			
0x000f	0x01	Position		0x01	deg	dm	grid	utm	mgrs		
0x0010	0x01	Card System	0x00-0xff	0xff (WGS84)							
0x0011	0x01	Grid		0x00 (M9) / 0x01-0x0a (X10 see table of local grids below)							
0x0012	0x01	UTM-MGRS Datum		0x00 (M9) / 0x01-0x0f (X10 see table of datums below)							
0x0013	0x01	???	(TM only)	0xff							

0x0014 cb 14 00 ... (M9)
 0x0014 89 12 ff ... (X10)

0x0017 0x01 Type of Watch
 0x5b 91 (M9)
 0x5e 94 (X10)

M9

0x0018 ff ff ff 59 32 0e 93 13 ...Y2...
 0x0020 ae 0e fc 13 42 16 c2 10B...
 0x0028 e2 16 70 11 70 0c 1f fd ..p.p...
 0x0030 51 0d 55 00 ac 0c 56 fd Q.U...V.
 0x0038 9d 24 4f fe 0c 04 60 24 ..\$O...`\$
 0x0040 18 12 32 f1 59 ff ..2.Y.

X10

0x0018 ff ff ff 59 e0 fa 72 76 ...Y..rv
 0x0020 00 00 d7 fe 2d bb fd ff-...
 0x0028 be a8 24 46 fc ff 68 50 ..\$.hP
 0x0030 b9 87 00 00 **bc 55 2b 6d**U+m
 0x0038 **0a 00 5a a9 0d 8d** 0b 00 ..Z.....
 0x0040 00 **00** 32 f1 59 ff ..2.Y.

Again X10 after storing new route with 5 way points

0x0018 ff ff ff 59 e0 fa 72 76 ...Y..rv
 0x0020 00 00 d7 fe 2d bb fd ff-...
 0x0028 be a8 24 46 fc ff 68 50 ..\$.hP
 0x0030 b9 87 00 00 **d0 51 ef 42**Q.B
 0x0038 **0b 00 11 f7 dc 90** 0b 00
 0x0040 00 **1c** 32 f1 59 ff ..2.Y.

0x0046 0x03 Menu Configuration (TM only - but works with M9)
 0x0046 0x01 Contrast 0x04 4
 0x0047 0x01 Light Mode 0x00 **normal** off night

0x0048	0x01	Brightness	0x03	3		
0x0049	0x02	Menu Configuration (TM & SM)				
0x0049	0x01	Tone	0x00	off	on	
0x004a	0x01	Info	0x01	off	on	
0x004b	0x03	Menu Configuration (TM only - but works with M9)				
0x004b	0x01	Sync	0x01	off	on	
0x004c	0x01	Declination	0x02	off	on	auto
0x004d	0x01	???	0x00	off	on	
0x004e	0x02	Timezone Info (TM & SM)				
0x004e	0x01	UTC + X * 15 minutes	0x08	(8 * 15 min = 2 hours)		
0x004f	0x01	???	0x20	32 (M9)		
			0x00	0 (X10)		

M9

0x0050 c9 64 92 52 8f 22 e1 dc .d.R."..
 0x0058 c5 57 f9 2e 59 ff ff ff .W..Y...

X10

0x0050 58 5b 28 55 aa 20 52 da X[(U. R.
 0x0058 9a 6d d4 2c 59 ff ff ff .m.,Y...

0x0060	0x08	GPS Rates (SM only - X10 responds with 8 Bytes 0xff)				
		M9 Fix Rates				
0x0060	0x02	Normal Fix Rate	0x01			
0x0061			0x00	0x0001	1 Second	
0x0062	0x02	Normal Save Rate	0x0a			
0x0063			0x00	0x000a	10 Seconds	
0x0064	0x02	Race Fix Rate	0x01			
0x0065			0x00	0x0001	1 Second	
0x0066	0x02	Race Save Rate	0x05			
0x0067			0x00	0x0005	5 Seconds	
0x0068	0x04	Serial Number ie. 12345678				
0x0068			0x0c	12		
0x0069			0x22	34		
0x006a			0x38	56		
0x006b			0x4e	78		

0x006c 0x0e GPS Rates (TM only - M9 responds with 14 Bytes * 0xff)

		X10 Fix Rates				
0x006c	0x02		0x01			
0x006d			0x00	0x0001	1 Second	
0x006e	0x02		0x01			
0x006f			0x00	0x0001	1 Second	
0x0070	0x02		0x0a			
0x0071			0x00	0x0010	10 Seconds	
0x0072	0x02		0x3c			
0x0073			0x00	0x003c	60 Seconds	
0x0074	0x02		0x3c			
0x0075			0x00	0x003c	60 Seconds	
0x0076	0x02		0x3c			
0x0077			0x00	0x003c	60 Seconds	
0x0078	0x02		0x3c			
0x0079			0x00	0x003c	60 Seconds	

0x0082	0x02	???	0x27, 0x28 (M9) - 0x27, 0xff (X10)
0x008c	0x07	???	0x08, 0x0e, 0x00, 0x00, 0x0f, 0x28, 0x01 (M9)
0x0093	0x01	???	0x01 (M9)
0x008c	0x07	???	0x08, 0x0e, 0x01, 0x00, 0x82, 0x28, 0x01 (X10)
0x0093	0x01	???	0x00 (X10 initial)
0x0093	0x08	???	0x39, 0x2e, 0x35, 0x2f, 0x34, 0x3a, 0x33, 0x32 (X10) "9.5/4:32"
0x009b	0x08	???	0x00, 0x00, 0xfc, 0x00, 0x60 (X10)

X10

0080	ff ff 27 ff ff ff ff	..!.....
0088	ff ff ff ff 08 0e 01 00
0090	82 28 00 32 36 2e 33 2f	.(.26.3/
0098	31 36 3a 32 37 00 00 dc	16:27...

X10

0x00a0 00 60 00 60 00 60 00 60 X10 Temperatures? – NO too hot!

0x00a8 00 60 00 60 00 60 00 60

0x00b0 00 60 00 60 00 60 00 60

0x00b8 00 60 00 5f 00 5f 00 ff

another example

0x00a0 ff f1 ff f1 ff f1 ff f1

0x00a8 ff f1 ff f1 ff f1 ff f1

0x00b0 ff f1 ff f1 ff f1 ff f1

0x00b8 ff f1 ff f1 ff f1 ff ff

and another

0x00a0 ff dc ff dc ff dc ff dc

0x00a8 ff dc ff dc ff dc ff dc

0x00b0 ff dc ff dc ff dc ff dc

0x00b8 ff dc ff dc ff dc ff ff

0x00c0 0x210 Barometer data (TM only - but works with M9) one value per word [hPa]

0x00c0 0xc0 Barometer data of the last 24 hours 96 words: 1 word per 15 minutes (0x00c0 = 0:00)

0x0180 0x150 Barometer data of the last 7 days 168 words: 1 word per hour (0x0180 = Monday 0:00)

0x2d0 0x0c ??? (X10)

M9

0x02d0 00 00 21 00 00 00 23 02 ..!...#.

0x02d8 01 08 29 00 03 27 31 04 ..)..!.

0x0352 0x01 Number of datums (X10 only - returns 0x07 | TM software can set 8 additional datums)

0x0353 0x60 Table of datums

Format of entries:

1 byte: ID

11 bytes: Name of entry

0x0353 0x0c 0, -----
 0x035f 0x0c 255, WGS84
 0x036b 0x0c 122, NAD83
 0x0377 0x0c 99, NAD27us
 0x0383 0x0c 107, NAD27ca
 0x038f 0x0c 102, ALASKA
 0x039b 0x0c 196, HAWAII
 0x03a7 0x0c 118, MEXICO

datums in TM software are:

WGS84
 NAD27
 W-USA
 E-USA
 ALASKA
 HAWAII
 NAS-E
 NAS-F
 NAS-G
 NAS-H
 NAS-I
 NAS-J
 MEXICO
 NAR-B
 NAR-C

0x04a2 0x02 Number of local grids (X10 only - M9 responds with 0xff)

0x04a4 0x78 Table of local grids (X10 only - M9 responds with mostly 0xff)

Format of entries:

1 byte: ID

11 bytes: Name of entry

0x04a4 0x0c Finnish
 0x04b0 0x0c Swedish
 0x04bc 0x0c British
 0x04c8 0x0c Swiss
 0x04d4 0x0c Irish
 0x04e0 0x0c NZTM
 0x04ec 0x0c Dutch
 0x04f8 0x0c BNM M28
 0x0504 0x0c BNM M31
 0x0510 0x0c BNM M34

0x0510 0x0c ??? (M9 only)

0x0510 fa 00 c5 64 00 dc 07 01 c5 42 01 c5 250, 50432, 100, 56320, 7, 50433, 66, 50433

0x0510 3f 00 c5 7e 00 c5 07 01 c5 42 01 c5 63, 50432, 126, 50432, 7, 50433, 66, 50433

0x0510 3e 00 c5 7e 00 c5 07 01 c5 42 01 c5 62, 50432, 126, 50432, 7, 50433, 66, 50433

0x05c0 ??? Table of way points ???

Format of entries:

1 word: way point ID

98 bytes: 0x00

end marker: way point ID == 0

X10

0x05c0 01 00 00 00 00 00 00 00	Table of Routes 1 (0x01) = MyPoints
0x05c8 00 00 00 00 00 00 00 00	
0x05d0 00 00 00 00 00 00 00 00	
0x05d8 00 00 00 00 00 00 00 00	
0x05e0 00 00 00 00 00 00 00 00	
0x05e8 00 00 00 00 00 00 00 00	
0x05f0 00 00 00 00 00 00 00 00	
0x05f8 00 00 00 00 00 00 00 00	
0x0600 00 00 00 00 00 00 00 00	
0x0608 00 00 00 00 00 00 00 00	
0x0610 00 00 00 00 00 00 00 00	
0x0618 00 00 00 00 00 00 00 00	
0x0620 00 00 00 00 ff ff ff ff	
0x0628 ff ff ff ff ff ff ff ff	
0x0630 ff ff ff ff ff ff ff ff	
0x0638 ea 01 00 00 00 00 00 00	490 (0x01ea) = Mempoind ID of Home
0x0640 00 00 00 00 00 00 00 00	
0x0648 00 00 00 00 00 00 00 00	
0x0650 00 00 00 00 00 00 00 00	
0x0658 00 00 00 00 00 00 00 00	
0x0660 00 00 00 00 00 00 00 00	
0x0668 00 00 00 00 00 00 00 00	
0x0670 00 00 00 00 00 00 00 00	
0x0678 00 00 00 00 00 00 00 00	
0x0680 00 00 00 00 00 00 00 00	
0x0688 00 00 00 00 00 00 00 00	
0x0690 00 00 00 00 00 00 00 00	
0x0698 00 00 00 00 ff ff ff ff	
0x06a0 ff ff ff ff ff ff ff ff	
0x06a8 ff ff ff ff ff ff ff ff	

0x06b0 32 00 01 00 01 00 00 00	2.....	50 (0x32), 1, 1 : 50 Routes, length 1 ID 1
0x06b8 4d 79 50 6f 69 6e 74 73	MyPoints	Name MyPoints
0x06c0 00 00 fc ff ff 00 00 00	
0x06c8 00 00 01 01 00 00 00 00	
0x06d0 ea 01 48 6f 6d 65 00 00	..Home..	490 (0x01ea) Home
0x06d8 00 00 00 00 00 00 ff ff	
0x06e0 ff ff 00 00 00 00 00 00	
0x06e8 00 00 00 02 ff ff ff ff	

Again X10

0x05c0 01 00 02 00 00 00 00 00	Table of Routes 1 (0x01) = MyPoints
0x05c8 00 00 00 00 00 00 00 00	
0x05c8 00 00 00 00 00 00 00 00	
0x05d0 00 00 00 00 00 00 00 00	
0x05d8 00 00 00 00 00 00 00 00	
0x05e0 00 00 00 00 00 00 00 00	
0x05e8 00 00 00 00 00 00 00 00	
0x05f0 00 00 00 00 00 00 00 00	
0x05f8 00 00 00 00 00 00 00 00	
0x0600 00 00 00 00 00 00 00 00	
0x0608 00 00 00 00 00 00 00 00	
0x0610 00 00 00 00 00 00 00 00	
0x0618 00 00 00 00 00 00 00 00	
0x0620 00 00 00 00 ff ff ff ff	
0x0628 ff ff ff ff ff ff ff ff	

```

0x0630 ff ff ff ff ff ff ff .....
0x0638 01 00 02 00 03 00 04 00 ..... Table of way points for last route
0x0640 05 00 00 00 00 00 00 00 .....
0x0648 00 00 00 00 00 00 00 00 .....
0x0650 00 00 00 00 00 00 00 00 .....
0x0658 00 00 00 00 00 00 00 00 .....
0x0660 00 00 00 00 00 00 00 00 .....
0x0668 00 00 00 00 00 00 00 00 .....
0x0670 00 00 00 00 00 00 00 00 .....
0x0678 00 00 00 00 00 00 00 00 .....
0x0680 00 00 00 00 00 00 00 00 .....
0x0688 00 00 00 00 00 00 00 00 .....
0x0690 00 00 00 00 00 00 00 00 .....
0x0698 00 00 00 00 ff ff ff ff .....
0x06a0 ff ff ff ff ff ff ff .....
0x06a8 ff ff ff ff ff ff ff .....
0x06b0 32 00 01 00 01 00 00 00 2.....
0x06b8 4d 79 50 6f 69 6e 74 73 MyPoints
0x06c0 00 00 fc ff ff 00 00 00 .....
0x06c8 00 00 01 01 00 00 00 00 .....
0x06d0 ea 01 48 6f 6d 65 00 00 ..Home..
0x06d8 00 00 00 00 00 00 00 ff .....
0x06e0 ff ff 00 00 00 00 00 00 .....
0x06e8 00 00 00 02 ff ff ff ff .....

```

```

0x06b0 0x00 ??? 0x00 (M9)

```

0x0d1e	0x01	Number of Mempoins (TM only - M9 responds with 0xff)	
0x0d20	0x0510	Table of Mempoins (TM only - M9 responds with 1296 Bytes * 0xff)	
		Format of entries:	
		18 (0x12) bytes per entry	
		3 words: local date and time binary coded in 6 bytes Y-M-D h:m:s	
		1 byte: Track ID	
		1 word: ??? (0x0000)	
		1 word: Altitude [m]	
		1 word: Heading [centi degrees]	
		1 word: Pressure [hPa]	
		1 byte: Temperature [C]	
		1 byte: Type of Mempoins [3 = start point, 0 = Markpoint, 1 = pause start point, 2 = pause end point]	
		1 byte: Mempoins Group of Markpoint (see Mempoins names below)	
0x0d20	0x12	1 st Entry	
0x0d38	0x12	2 nd Entry	
0x0d50	0x12	3 rd Entry	
...			
0x1218	0x12	54 th Entry	
0x1500	0x16	???	4f 4b 4f 4b c1 37 7a 4b d1 51 f7 5f 12 52 12 52 01 00 00 00 b9 02 (M9) O K O K 7 z K Q _ R R 79 75 79 75 193 55 122 75 209 81 247 95 18 82 18 82 1 0 0 0 185 2
0x1fff	0x01	0x5a	???
0x2000	0x01	0x00	???
0x1fea	0x01	Number of Mempoins names (TM only - M9 responds with 0xff - X10 responds with 0x13)	
0x1feb	0xf0	Table of Mempoins names (TM only - M9 responds with 240 bytes most of them 0xff)	
		X10 responds with table of Mempoins names:	
		Format of entries:	
		1 Byte:	ID
		11 Bytes:	Name including delimiting 0x00s
0x1feb	0x0c	0x00	River
0x1ff7	0x0c	0x01	Hill
0x2003	0x0c	0x02	Mountain
0x200f	0x0c	0x03	Cross
0x201b	0x0c	0x04	Rock
0x2027	0x0c	0x05	Begin
0x2033	0x0c	0x06	End
0x203f	0x0c	0x07	Valley
0x204b	0x0c	0x08	Creek
0x2057	0x0c	0x09	Clift
0x2063	0x0c	0x0a	Canyon
0x206f	0x0c	0x0b	Road
0x207b	0x0c	0x0c	Trail
0x2087	0x0c	0x0d	Forest
0x2093	0x0c	0x0e	Meadow
0x209f	0x0c	0x0f	Spring
0x20ab	0x0c	0x10	Cave
0x20b7	0x0c	0x11	Coast
0x20c3	0x0c	0x12	Camp
0x20cf	0x0c	0x13	Hiking

0x20e3	0x02	1 st byte	next free altitude profile page	(X10)
		2 nd byte	???	(X10)
0x20e5	0x19	1 st to 25 th byte	Table of track header IDs	(TM only - M9 responds with 25 bytes * 0xff) X10 returns 25 bytes (track IDs)
0x2100	0x0640	Table of track header additional information (TM only - M9 responds with 0xffs)		
		Format of entries:		
		33 (0x21) bytes		
		1 byte:	page no. of altitude profile data	
		3 words:	local date and time binary coded in 6 bytes Y-M-D h:m:s	
		1 byte:	10, 20, 60 (0x0a, 0x14, 0x3c) altitude profile interval [s]	
		1 word:	ascend in [m]	
		1 word:	descend [m]	
		1 word:	# rounds	
		2 words:	duration in [s]	
		1 word:	maximum altitude [m]	
		1 word:	minimum altitude [m]	
		1 word:	track ID	
		1 byte:	# Mempoins	
		1 word:	average speed [cm/s]	
		1 word:	maximum speed [cm/s]	
		2 words:	distance [m]	
0x2100	0x21	1 st Entry		
0x2140	0x21	2 nd Entry		
0x2180	0x21	3 rd Entry		
...				
0x2700	0x21	25 th Entry		
0x2780	0x5680	174 Pages of altitude profile data (TM/X10 only - M9 responds with mostly 0xffs)		
		Format of entries:		
		n bytes		
		(n) bytes:	altitude delta [m] to previous interval altitude	
			or tag followed by tag special data	
Possible values for tags are:				
	0x7c	absolute altitude		
		followed by		
		1 word:	start pressure [hpa]	
		1 byte:	start altitude [m] (without offset)	
	0x7d	change profile interval		
		followed by		
		1 byte:	new interval [s]	
	0x7e	altitude offset		
		followed by		
		1 word:	altitude offset [m]	
	0x80	end of data		
0x2780	0x80	1 st page		
0x2800	0x80	2 nd page		
0x2880	0x80	3 rd page		
...				
0x7d80	0x80	174 th page		
0x2ec0	0x02	???	0x00, 0x2b : 0, 43 (M9)	
0x2eeb	0x03	???	0x0c, 0x00, 0x2c : 12, 0, 44 (M9)	

0x2ef5	0x03	???	0x0c, 0x00, 0x1d : 12, 0, 29	(M9)
0x2eff	0x01	???	0x0c : 12	(M9)
0x2f09	0x03	???	0x0c, 0x00, 0x1c : 12, 0, 28	(M9)
0x3fff	0x01	???	0xa5	(M9)
0x4000	0x02	???	0x00, 0x00	(M9)
0x4700	0x64	???	0x01, 0x00, 0x02, 0x00, 0x03, 0x00, 0x30, rest 0x00	(M9 only)
0x4700	0x64	???	0x01, rest 0x00	(M9)

0x5200 0x0320 Table of track header additional information (SM only - X10 responds with altitude profile data)

Format of entries:

25 (0x19) bytes = 12 words + 1 byte

1 word: track ID

3 words: local date and time binary coded in 6 bytes Y-M-D h:m:s

1 word: ???

2 words: distance [m]

2 words: duration [s]

1 word: maximum speed [cm/s]

1 word: average speed [cm/s]

1 word: # Mempoins

1 byte: 0 ???

0x5200 0x19 1st Entry

0x5220 0x19 2nd Entry

0x5240 0x19 3rd Entry

...

0x5500 0x19 25th Entry

0x6000 0x01 Number of Mempoins (including Windpoints) 1 byte (M9)

0x6020 0x0320 Table of Mempoins (M9 only)

Format of entries:

22 (0x16) bytes per entry

3 words local date and time binary coded in 6 bytes Y-M-D h:m:s

1 word: Trackpoint ID

1 word: ??? (0x0000)

1 word: (Wind) direction (bearing) [degrees]

1 word: ??? (0x0000)

1 word: Pressure [hPa]

1 byte: Temperature [C]

1 word: ??? (0x0000)

1 byte: Wind speed

1 byte: Type of Mempoins [0 = Mempoins, 1 = Windpoint, ... ???]

1 byte: Track ID

0x6020 0x16 1st Entry

0x6040 0x16 2nd Entry

0x6060 0x16 3rd Entry

...

0x6320 0x16 25th Entry

0x7920 0x01 ??? 0x00 (M9)

0x7b00 0x190 Round robin table of timestamps with one or two values??? (what the heck is this???) (M9)

Format of entries:

3 words local date and time binary coded in 6 bytes Y-M-D h:m:s

1 word or 2 bytes: Don't have any glue what they mean.

Don't even have a clue what rule is behind the dates.

Are they GPS on/off times???

X10

0x7fe0 ff ff ff ff ff 01 300

0x7fe8 34 5f 31 36 30 36 31 30 4 160610

0x7ff0 31 35 33 36 31 38 00 00 153618..

0x7ff8 02 ff ff ff ff ff ff

Where are the following settings stored ???

Dual Time

Barometer Alarm off on 0x00 ???

Declination data East / West, Declination in 0.5° Steps

Where are the temperature data in the X9, X9i and X10 stored?

Examples of Memory Locations

```
C:\...\Visual Studio 2008\Projects\suunto-cli\Debug>suunto-cli.exe com4 0x0000 0x60
```

```
0000 01 ff ff ff 28 ff 01 01 00 00 00 02 03 02 03 01
0010 ff 01 01 ff cb 14 00 5b ff ff ff 59 32 0e 93 13
0020 ae 0e fc 13 42 16 c2 10 e2 16 70 11 31 0b cd fe
0030 79 0e 96 01 92 0d 7a 00 9e 24 4f fe 0c 04 61 24
0040 18 1d 32 f1 59 ff 04 00 03 00 00 01 02 00 04 20
0050 c9 64 92 52 8f 22 e1 dc c5 57 f9 2e 59 ff ff ff
```

```
C:\...\Visual Studio 2008\Projects\suunto-cli\Debug>suunto-cli.exe com4 0x0060 0x60
```

```
0060 01 00 0a 00 01 00 01 00 0c 22 38 4c ff ff ff ff
0070 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
0080 ff ff 27 28 ff ff ff ff ff ff ff ff 08 0e 00 00
0090 0f 28 01 01 ff ff ff ff ff ff ff ff ff ff ff
00a0 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
00b0 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
```

```
C:\...\Visual Studio 2008\Projects\suunto-cli\Debug>suunto-cli.exe com4 0x00c0 0x240
```

```
00c0 c1 03 c1 03 c1 03 c1 03 c1 03 c1 03 c1 03
00d0 c1 03 c1 03 c1 03 c1 03 c0 03 c0 03 c0 03 c0 03
00e0 c0 03 c0 03 c0 03 c0 03 c0 03 c0 03 bf 03 bf 03
00f0 bf 03 bf 03 bf 03 bf 03 be 03 bf 03 bf 03 bf 03
0100 bf 03 bf 03 be 03 bf 03 bf 03 bf 03 be 03 be 03
0110 be 03 be 03 bd 03 be 03 bd 03 bd 03 bd 03 bd 03
0120 bd 03 bc 03 bd 03 bc 03 ba 03 ba 03 b8 03 ba 03
0130 c3 03 b9 03 b9 03 b9 03 b8 03 ca 03 ca 03 ca 03
0140 ca 03 ca 03 ca 03 ca 03 ca 03 ca 03 ca 03 ca 03
0150 ca 03 ca 03 ca 03 ca 03 ca 03 c9 03 c9 03 c9 03
0160 ca 03 c3 03 c2 03 c3 03 c3 03 c3 03 c3 03 c3 03
0170 c3 03 c3 03 c3 03 c2 03 c2 03 c2 03 c2 03 c2 03
0180 bd 03 bc 03 bd 03 bc 03 bc 03 bc 03 bc 03 bd 03
0190 be 03 c7 03 c7 03 c7 03 c7 03 c7 03 c7 03 c7 03
01a0 c7 03 c8 03 c8 03 c0 03 c0 03 c0 03 c1 03 c1 03
01b0 c1 03 c0 03 c0 03 c0 03 c0 03 c0 03 c0 03 c0 03
01c0 c1 03 ca 03 ca 03 cb 03 ca 03 c9 03 c9 03 c9 03
01d0 c9 03 c9 03 c9 03 c1 03 c1 03 c1 03 c1 03 c0 03
01e0 c0 03 bf 03 bf 03 bf 03 bf 03 bf 03 bf 03 bf 03
01f0 bf 03 c8 03 c7 03 c8 03 c7 03 c7 03 c7 03 c7 03
0200 c6 03 c6 03 c6 03 c6 03 be 03 be 03 be 03 be 03
0210 bd 03 bd 03 bd 03 bd 03 bd 03 bd 03 bd 03 be 03
0220 be 03 c8 03 c8 03 c8 03 c9 03 c8 03 c8 03 c9 03
0230 c9 03 c9 03 ca 03 c1 03 c2 03 c2 03 c3 03 c2 03
0240 c3 03 c2 03 c2 03 c2 03 c2 03 c2 03 c2 03 c3 03
0250 c4 03 cd 03 cc 03 cd 03 cc 03 cc 03 cd 03 cc 03
0260 cc 03 cd 03 c4 03 c4 03 c4 03 c4 03 c4 03 c4 03
0270 c4 03 c4 03 c3 03 c3 03 c3 03 c3 03 c3 03 c4 03
0280 c4 03 c3 03 c3 03 c3 03 c3 03 c4 03 c3 03 ca 03
0290 ca 03 ca 03 ca 03 ca 03 ca 03 c3 03 c3 03 c2 03
02a0 c1 03 c1 03 c1 03 c0 03 c0 03 c0 03 bf 03 be 03
02b0 bf 03 bf 03 be 03 bd 03 bd 03 ba 03 c3 03 b8 03
02c0 bc 03 bc 03 bd 03 bd 03 bd 03 bd 03 bd 03 bd 03
02d0 05 2d 3b 04 00 00 01 03 ff ff ff ff ff ff ff ff
02e0 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
02f0 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
```

Examples of Annotated Protocol Tracelogs

First Part of communication between Trek/Sail Manager and M9
Trek Manager displays error message: Device not X9!M9

Trek and Sail Manager just have different filling bytes and
therefore different checksums.
But the workload data is identical!!!

```
Read Memory 1 Byte at Address 0x0017 (Type of Watch)
[02/10/2009 10:57:03] - Written data
3c 2a "<*" Record start
00 0d    13 Record data words
ec f6 Reserved should be 0x0000
12 00 Reserved should be 0x0000
23 3e Reserved should be 0x0000
03 f3 Old Suunto D6/D9 protocol
00 04 wMsgNumber should be 0x0000
02 0c wSource = Host node (0x02) / Archive (0x0c)
01 07 wDest = iTrax node (0x01) / Custom Serial Protocol out (0x07)
00 01 nTrId = Transaction Id
00 04    4 iTalk Datenworte
05      Read Memory
00 03 Parameter length 3 Bytes
00 17 Address = 0x0017
01      Length = 1 Byte
10      XOR Checksum of Read Memory
00      Random Value
2f 01 Checksum of 4 iTalk data words = 0x0500 + 0x0300 + 0x1701 + 0x1000
3e      ">" Record end
```

```
[02/10/2009 10:57:03] - Read data
3c 2a "<*" Record start
00 0a    10 Record data words
26 07 Reserved is arbitrary
26 06 Reserved is arbitrary
00 00 Reserved is arbitrary
03 f3 Old Suunto D6/D9 protocol
00 01 wMsgNumber is arbitrary
01 07 wSource = iTrax node (0x01) / Custom Serial Protocol out (0x07)
02 0c wDest = Host node (0x02) / Archive (0x0c)
ff ff nTrId = Transaction Id = -0x0001
00 01    1 iTalk data word
80 00 ACK (OK)
80 00 Checksum of 1 iTalk data word = 80 00
3e      ">" Record end
```

```
3c 2a "<*" Record start
00 0e    14 Record data words
26 07 Reserved is arbitrary
26 06 Reserved is arbitrary
00 00 Reserved is arbitrary
03 f3 Old Suunto D6/D9 protocol
00 05 wMsgNumber is arbitrary
01 07 wSource = iTrax node (0x01) / Custom Serial Protocol out (0x07)
02 05 wDest = Host node (0x02) / Custom Serial Protocol in (0x05)
00 00 nTrId = Transaction Id
00 05    5 iTalk data words
05      Read Memory
00 05 Parameter Length 5 Bytes
00 17 Adresse = 0x0017
01      Length 1 Byte
5b      read Byte 0x5b = 91 = M9
ff      filler
c3      filler
71      XOR of Read Memory (incl. fillers)
40 71 Checksum of 5 iTalk data words = 0x0500 + 0x0500 + 0x1701 + 0x5bff + 0xc371
3e      ">" Record end
```

 Get Memory Usage (wmsgtype = 0x03f4)

```
[02/10/2009 10:49:14] - Written data
3c 2a 00 09 68 f9 12 00 2d 52 03 f4 00 00 02 0c <*.hù..-R.ó....
01 07 00 2d 00 00 00 00 3e ..-....>

[02/10/2009 10:49:14] - Read data
3c 2a 00 0a 26 07 26 06 00 00 03 f4 00 01 01 07 <*.&.&....ó....
02 0c ff d3 00 01 00 62 00 62 3e ..ÿÓ...b.b> # Free Memory = 98 %
```

 Get Type of Watch

```
Read Memory Address 0x0017 0x01 Bytes
[02/10/2009 10:49:14] - Written data
3c 2a 00 0d 70 ec 12 00 23 3e 03 f3 00 04 02 0c <*.pi..#>.ó....
01 07 00 2e 00 04 05 00 03 00 17 01 10 00 2f 01 ...../. # Read Address 0x0017
3e >

[02/10/2009 10:49:14] - Read data
3c 2a 00 0a 26 07 26 06 00 00 03 f3 00 01 01 07 <*.&.&....ó....
02 0c ff d2 00 01 80 00 80 00 3e 3c 2a 00 0e 26 ..ÿÓ...e.e.><*.&
07 26 06 00 00 03 f3 00 05 01 07 02 05 00 00 00 .&....ó.....
05 05 00 05 00 17 01 5b ff ff 4d 7c 4d 3e .....[ÿÿM|M> # Byte at Address 0x0017 =
0x5b = 91 = M9
```

 Get Barometer History Data from Suunto M9:

In 2 Steps

- 1) Get 96 data sets (96 words = 192 Bytes) for the last 24 hours. 1 set every 15 minutes.
- 2) Get 168 data sets (168 words = 336 Bytes) for the last 7 days. 1 set every hour.

```
Read Memory 0x64 (100) Bytes from Address 0x00c0
[02/10/2009 10:49:14] - Written data to M9
3c 2a "<*" Start of Record
00 0d 13 Record data words
10 ed Reserved should be 0x0000
12 00 Reserved should be 0x0000
23 3e Reserved should be 0x0000
03 f3 Old Suunto D6/D9 protocol
00 04 wMsgNumber should be 0x0000
02 0c wSource = Host node (0x02) / Archive (0x0c)
01 07 wDest = iTrax node (0x01) / Custom Serial Protocol out (0x07)
00 2f nTrId = Transaction Id
00 04 4 iTalk data words
05 Read Memory
00 03 Parameter length = 3 Bytes
00 c0 Address 0x00c0
64 Length 0x64 = 100 Bytes
a2 XOR Checksum of Read Memory
00
6a 64 Checksum of 4 iTalk data words
3e ">" End of Record
```

Return ACK

Return 100 Bytes

```
[02/10/2009 10:49:14] - Read data from M9
3c 2a "<*" Start of Record
00 0a 10 Record data words
26 07 Reserved is arbitrary
26 06 Reserved is arbitrary
00 00 Reserved is arbitrary
03 f3 Old Suunto D6/D9 protocol
00 01 wMsgNumber is arbitrary
01 07 wSource = iTrax node (0x01) / Custom Serial Protocol out (0x07)
02 0c wDest = Host node (0x02) / Archive (0x0c)
ff d1 nTrId = Transaction Id -0x002f
00 01 1 iTalk data words
80 00 ACK (OK)
80 00 Checksum of 1 iTalk data word = 80 00
3e ">" End of Record
```



```

03 c4 03 c4 03 cc 03 cc 03 cc 03 cb 03 cb 03 cb      .Ä.Ä.Ï.Ï.Ï.Ë.Ë.Ë
03 ca 03 ca 03 c2 03 c3 03 c3 03 c3 03 c3 03 c3      .Ë.Ë.Ä.Ä.Ä.Ä.Ä.Ä
03 c3 03 c3 03 c3 03 c2 03 c2 03 c2 03 c2 03 c3      .Ä.Ä.Ä.Ä.Ä.Ä.Ä.Ä
03 c3 03 c4 03 c5 03 c5 03 c3 03 c3 03 ca 03 ca      .Ä.Ä.Ä.Ä.Ä.Ä.Ë.Ë
03 ca 03 ca 03 ca 03 ca 03 cd 03 c5 03 c4 03 c3      .Ë.Ë.Ë.Ë.Ï.Ä.Ä.Ä
03 c3 03 c3 03 c3 03 c3 03 c3 03 c3 79 44 97 40 3e  .Ä.Ä.Ä.Ä.Ä.yD-@>
    
```

```

Get Barometer History Data from suunto m9:
Read Memory 0x24 (36) Bytes from Address 0x02ac
[02/10/2009 10:49:17] - Written data to M9
3c 2a 00 0d 10 ed 12 00 23 3e 03 f3 00 04 02 0c      <*.ï.ï.#>.ó....
01 07 00 34 00 04 05 00 03 02 ac 24 8c 00 40 26      ...4.....-S@E.@&
3e                                                       >
    
```

```

Return ACK
Return 36 Bytes
[02/10/2009 10:49:17] - Read data from M9
3c 2a 00 0a 26 07 26 06 00 00 03 f3 00 01 01 07      <*...&.&....ó....
02 0c ff cc 00 01 80 00 80 00 3e 3c 2a 00 1f 26      ..ÿÏ...€€.><*...&
07 26 06 00 00 03 f3 00 16 01 07 02 05 00 00 00      .&....ó.....
16 05 00 27 02 ac 24 c3 03 c3 03 c3 03 c3 03 c4      ...'..-SÄ.Ä.Ä.Ä.Ä
03 c4 03 c4 03 c4 03 e2 03 cd 03 cd 03 cd 03 cd      .Ä.Ä.Ä.ä.Ï.Ï.Ï.Ï
03 cd 03 cd 03 ce 03 ce 03 ce 03 c3 47 d1 a3 3e      .Ï.Ï.Ï.Ï.Ï.ÄGÑE>
    
```