

s725xdump - dumping Polar S725X data in Linux

After a weekend of digging through hex dumps, I managed to produce a little tool that dumps graphs from the Polar S725X bicycle monitor slash exercise watch in Linux. I've only tested it on my own device with my own settings, but I would guess it works on at least the S625X too.

It's rather meager, but it works well enough to dump the heart rate logs in a gnuplotable format. It will show timestamps, durations, kcals and lap times. I also have no idea about altitude, distances, etc.

— [Vidar Hølen](#). [More junk](#). [Mail Vidar](#).

Usage

[Get it here](#). Feel free to plunder if you want to make a more sturdy and polished app. There's also [a contributed patch](#) to make it a bit more intelligent.

To run:

```
$ make s725xdump
$ ./s725xdump > dumpfile
```

Make sure that your system supports IrDA first. The drivers should be loaded, and the irda0 interface should exist and be UP.

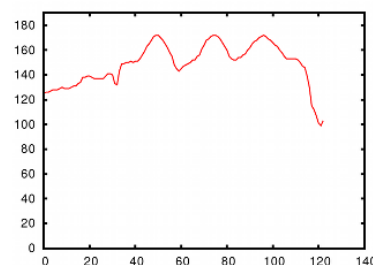
Sample run

This is what it should look like:

```
vidar@kelvin ~/c/irda $ ./s725xdump > dumpfile
Doing device discovery.....
Discovered: (list len=1)
  name: Polar S725X
  daddr: ec40565b
  saddr: d8baa0e0

Connected!
Device time is 02:31:39, 14/01 2007
In 3
In 2
In 1
Done downloading

vidar@kelvin ~/c/irda $ cat dumpfile
# Date: 2007-01-13 23:46:55
# Duration: 00:02:04.1
# AvgHR: 55
# MaxHR: 63
# Laps: 4
# RecordingRate: 5 secs
## Number   Heartrate Unknown Unknown
0   0 0 0
1   0 0 0
2   60 0 0
3   55 0 0
4   55 0 0
5   56 0 0
6   56 0 0
7   54 0 0
(etc)
```



Partial specification of the S725X protocol

Connecting

Connect to the device over an IR socket using "HRM" as service name.

Commands

Most of the commands I used were single-byte. Most of this is specified in hex. All data is big-endian except where indicated.

0x02 — Get device time

Send the byte 02, and you get a response like

```
02      #Responding to command 02
06 54 01 #Timestamp, ss:mm:hh (BCD), in this case 01:54:06
14 07    #date/year, 30. [20]04
61      #Lower four bits, Month (January)

00 10 44 10 00 #Unknown
```

0x09 — Reset device

Send the byte 09, and you get no response but the device is fairly completely reset.

0x0B — Transfer all files

Send the byte 0B, and you will receive a packet. Then send "16 2F" to receive more. The packet structure is like this:

```
0b # Response to command 0B
80 01 # Packets remaining. If top bit is set, like here, this is the first packet. There is 1 more.
(data)
```

Continue to send "16 2F" until you receive "16 02". This means that there is no more data. The (data) sections, when concatenated, are like this:

```
00 b0 # The number of following bytes, not including this one nor the next two
26 08 # Unknown but constant, probably device model. Not included in byte count.

#File (the rest of this page, down to and including heart rate samples, is repeated once per file)
b0 00 # The number of bytes in this file, little-endian (including this number). 176 dec in this case
00 00 00 00 # Unknown, presumably reserved
00 00 00 00
40 54 10 # Timestamp, BCD. 10:54:40
30 04 # 30. of the month, [20]04
9a # Lowest four bits: month (October)
# Highest four bits: tenth of a second (.9)
59 00 00 # Duration, BCD. 00:00:59.9

3e # Average Heartrate (62)
43 # Maximum heartrate (67)
01 # Number of laps
01 # Unknown, same as number of laps

00 00 00 20 # Unknown
00 #Recording interval: 0, 1 or 2 for 5, 15 or 60 secs

01 50 a0 50 a0 50 a0 00 01 fb 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 #37 dec bytes Unknown watch configuration options

01 #Best lap number
90 59 00 00#Best lap time (0:00:59.9)

00 00 00 #Unknown
00 00 #Kcal (BCD, little-endian). 0 here, but another example is 82 03 meaning 382 dec

00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 80 00 80 00 80 00 00 00 00 00 00 00 00 00 64 #34 bytes unknown

-- (Packet breaks aren't important, but for reference, there's one here)

# 21 dec bytes Unknown
50 a0 59 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 be

# 10 dec bytes of Lap information. Repeat as many times as necessary.
7b # Start of lap, seconds: 01 111011 (low 2 bits of fraction, 59 sec)
80 # Start of lap, minutes: 10 000000 (high 2 bits of fraction, 0 min)
00 # Hours. This lap started after 0:00:59.9
43 3e 43 # Heart rates
00 00 00 00 # Reserved?

#The rest of the bytes in the file (I think you have to keep track) are samples:
#Note, the last recorded sample comes first!
42 00 00 #Heartrate, unknown, unknown
42 00 00
3e 00 00
3f 00 00
3c 00 00
3b 00 00
41 00 00
3e 00 00
00 00 00
00 00 00
00 00 00
```