iRiver s10

This is I have to say a rather nice little music device. It's plays oggs which is of course fantastic, the only drawback is the strange playlist '.pla' format. These can be written using the windows only software which comes with the player, but for those of us on linux there is no such facility. I have spent some time with a hex editor reading the playlist file generated by the player and have determined the format which I detail here.

The file can be considered to be broken up into 512 byte chunks, the first is the header and every one after that is a track.

File Header Format

#

Error rendering macro 'column'

com.atlassian.renderer.v2.macro.basic.validator.MacroParameterValidationException: Width parameter must be a number (optionally followed by 'px' or '%').

Error rendering macro 'column'

com.atlassian.renderer.v2.macro.basic.validator.MacroParameterValidationException: Width parameter must be a number (optionally followed by 'px' or '%').

The first 4 bytes are the number of tracks in the file, this appears to be bigendian as I have only seen the 4th of these bytes with anything in it.

The next 28 bytes are the format header, this is *"iriver UMS PLA"* followed by 14 zeroed bytes.

The remaining 480 bytes contain the playlist heading. A quick list generated on the player will contain *"Quick List"* with the rest filled with zeroed bytes. The s10 will actually list playlists by filename, so what you have written here isn't as far as I'm aware used for anything.

Song Format

The track listings appear to be in bigendian 16-bit format. That is there are two bytes for each character but I have only seen the second of these used for anything. I'm assuming the extra space is used for non-latin character sets but all the lists I've been working with simply have a *'\0'* as every other byte.

The first 2 bytes specify the 16-bit byte in this chunk from which to read the display part of the filename, that being the first character after the path. The remainder of the 512 bytes contain the path and file name of the track relative to the root of the s10 with '\' slashes.

So for example to map to the file
`/Music/wish you were here/01 - shine on you crazy diamond part one.ogg`
we would get (in hex)

27	\	M	u	s	i	c	\	w	i	s	h	
00 1B	00 5C	00 4D	00 75	00 73	00 69	00 63	00 5C	00 77	00 69	00 73	00 68	00 20

y	o	u		w	e	r	e		h	e	r	e	\
00 79	00 6F	00 75	00 20	00 77	00 65	00 72	00 65	00 20	00 68	00 65	00 72	00 65	00 5C

0	1		-		s	h	i	n	e		o	n		y	o	u	
00 03	00 31	00 20	00 2D	00 20	00 73	00 68	00 69	00 6E	00 65	00 20	00 6F	00 6E	00 20	00 79	00 6F	00 75	00 20

c	r	a	z	y		d	i	a	m	o	n	d	
00 63	00 72	00 61	00 7A	00 79	00 20	00 64	00 69	00 61	00 6D	00 6F	00 6E	00 64	00 20

p	a	r	t		o	n	e	.	o	g	g
00 70	00 61	00 72	00 74	00 20	00 6F	00 6E	00 65	00 2E	00 6F	00 67	00 67

The rest of the 512 byte chunk should be filled with zeros.

Writing a List

A python script can be found here which does a decent job. It gets a couple of things wrong, namely the number of tracks before the header and the first 2 bytes of the track listing, so an incorrect track name will be displayed.

I have updated it to correct these two issues, iRiverPlaMaker.py

To continue with the example, the way to use the script would be from within the `/Music/wish you were here` directory on the s10.
`# ls | ../../iRiverPlaMaker.py ../../Playlists/wish\ you\ were\ here.pla`
when the python script is in the root directory of the s10.