RSP6 ticket format

https://git.eta.st/eta/rsp6-decoder

Bit start *Bit end*	Length	Datatype	Description	Notes	
0 - 1	1	bool	manual inspection required		
1 - 8	7	**unknown**		*many different values, meaning unknown*	
8 - 62	54	string	ticket reference		
62 - 68	6	string	checksum character		
68 - 72	4	int	version number		
72 - 73	1	bool	is standard class ticket		
73 - 91	18	string	Lennon ticket type		
91 - 109	18	string	fare label	as per the fares data	
109 - 133	24	string	origin	4-digit NLC	
133 - 157	24	string	destination	4-digit NLC	
157 - 181	24	string	retailer	4-digit NLC	
181 - 182	1	bool	is child ticket		
182 - 184	2	int	coupon type	[0, 1, 2, 3] ⇛ (single, season, return-outbound, return-inbound)	
184 - 194	10	int	discount code	as per the fares data	
194 - 211	17	int	route code	as per the fares data	
211 - 225	14	int	start time, days	since 1997-01-01	
225 - 236	11	int	start time, seconds		
236 - 238	2	int	depart time flag	[0, 1, 2, 3] ⇛ (none, ???, Specific, Suggested)	
238 - 255	17	int	passenger ID	*see Rust code for how to decode, but this isn't really used*	
255 - 327	72	string	passenger name		
327 - 329	2	int	passenger gender	*don't know what the mappings are; never seen this!*	
329 - 347	18	string	restriction code		
347 - 371	24	string	***Unknown*** — some NLC?	*but encoded differently to normal chars; LNER tix have this*	
371 - 372	1	**unknown**		*usually zero*	
372 - 373	1	bool	bidirectional		
379 - 383	4	int	limited duration code	*see Rust code for lookup table; old-style mTickets only?*	
383 - 384	1	bool	free text is extended		
384 - 385	1	bool	is full ticket		
385 - 386	1	bool	contains free text		
386 - 390	4	int	number of reservations		
390 - 404	14	int	purchase time, days	since 1997-01-01	**PURCHASE DETAILS** — only present if "is full ticket" is set (shift below bits up if not)
404 - 415	11	int	purchase time, seconds		
415 - 436	21	int	price (pence)		
449 - 497	48	string	purchase reference		
497 - 506	9	int	days of validity	zero in this field means "1 day only"	
506 - 512	6	***unknown***		*always zero so far*	
512 - 518	6	char	reserved service ID, first char		**RESERVATIONS** — repeated as indicated under "number of reservations" (can be none)
518 - 524	6	char	reserved service ID, second char		
524 - 538	14	int	reserved service ID, numbers		
538 - 544	6	char	reserved coach letter		
544 - 550	6	char	reserved seat letter		
550 - 557	7	int	reserved seat number		
557 - 783	226	string	free text		**FREE TEXT** — if "contains free text" is set, starts where reservations end until bit 783 or 863, depending on "free text is extended"
779 - 863	84	string	extended free text		
863 - 928	65	***unknown***			

General notes on decoding

Treat the ticket, after base26-decoding, as one long big-endian bitfield.
Indices and lengths are all measured in bits.

booleans are either 0 (false) or 1 (true).
integers are big-endian integers of the size specified.
strings and chars are composed of 6-bit wide integers; add 32 to the integer value to get an ASCII character code.

Time is specified as a number of days since 1997-01-01 (the year of privatisation!), and seconds since midnight.

I'm sorry this is bad I made it in LibreOffice okay