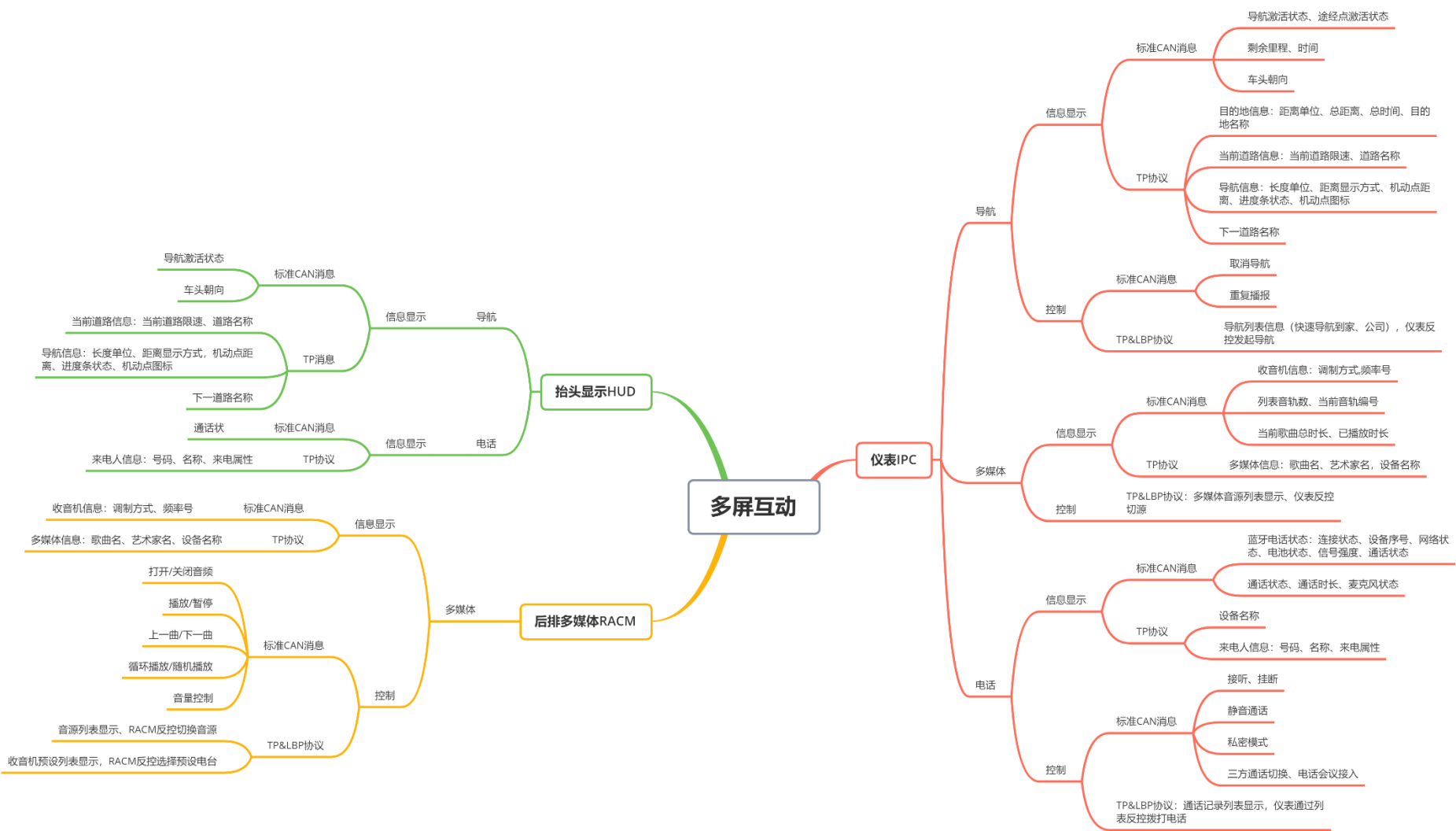


概要

车载中控娱乐系统会通过自身的触摸大屏为用户提供导航、电话和多媒体娱乐三大核心服务，但从整车系统设计角度考虑某些应用场景下中控娱乐大屏并不是用户获取以上信息最便捷的途径，例如对于正在开车的驾驶员而言从驾驶安全角度考虑仪表或 HUD 抬头显示会是获取以上信息较方便的途径，二对于第二排乘客而言后排的多媒体触摸屏会是获取多媒体娱乐信息较方便的途径。所以以上应用场景要求中控娱乐系统需要具备将这三个核心服务的信息能够共享到仪表、HUD 和后排娱乐屏幕的能力，同时娱乐系统也需要具备一定的接受这三个外部 ECU 反向控制的能力，以上能力综述为多屏互动。



多屏互动相关文档

基础协议

Transport Protocol APIM SPSS v1.19 Feb 27, 2019.pdf

该文档主要定义基于 ISO-15765-2 协议福特自行定义封装的连续帧协议 (TP)，文档主要定义了协议的基本数据格式、应答机制，以及具体的每条软件逻辑信号中个字段的数据格式和用途。多屏互动里面需要传输字符串信息、列表信息或者复杂图标信息时需要用到 TP 协议来传输这些信息。

List Browser Protocol APIM SPSS v1.8.1 Jan 23, 2020.pdf

该文档为列表协议，LBP 协议是在已封装好的 TP 协议基础上针对列表信息的传输和反向控制进一步封装的协议。该协议会定义基本的列表请求和回复应答机制，各类列表的属性、包含字段和具体数据结构。实际应用中当中控需要对外交互音源列表、快速导航列表和电话列表时需要用到 LBP 协议。

相关应用文档

BT Connectivity APIM SPSS v3.0 Feb 14, 2019.pdf

该文档为蓝牙连接规范文档，其中会详细定义蓝牙连接后名中控需要发送那些信号及详细的发送时序，来对外输出对应蓝牙设备状态、来电信息和通讯录信息。

NAV Repeater V2 APIM SPSS v1.2 Nov 4 2016.pdf

该文档具体定义了导航相关的信息如何通过中控传输到外部 ECU。

Rear Audio Control APIM SPSS v1.4 Jul 1, 2017.pdf

该文档具体定义了后排多媒体控制屏和中控交互的相关逻辑

多屏互动联系人

蓝牙：Xu, Amy (X.) XYAFEl@ford.com, Xu, Xin (X.) xxu91@ford.com,

多媒体：Xiang, Zhengxi (Z.) ZXIANG6@ford.com,

音频管理：Xia, Meng (M.) mxia13@ford.com,

仪表：Cao, Helen (H.) hcao22@ford.com, Li, Eureka (Y.) yli326@ford.com,

HUD：Qiu, Nao (N.) nqiu2@ford.com,

RACM：Xie, Will (W.) wxie15@ford.com, FNA RAM D&R：Pasupuleti, Kota (K.) kpasupu1@ford.com,

其余 Spec 的问题可以直接找 Spec 的 Author 联系。

多屏互动相关信号

对应信号可以参考之前整理的 Interaction.xlsx 文档，里面详细归纳了对应信号的用途，信号名称、message 名称和 Message ID。

多屏互动知识点

1. TP 连续帧协议

福特的 TP 连续帧协议是基于 ISO-15765-2 标准协议的基础，对相应的连续帧数据位做进一步封装。

下图是标准 TP 帧的帧结构 (摘自 15765 标准)，标准帧里的首帧的第一个 Byte 是以 0x10 开头，第二个 Byte 代表该组连续帧的数据长度，从第三个 Byte 开始是标准数据位。从第二帧开始，每一帧都是以 0x2X 开头 (X=1~F)。

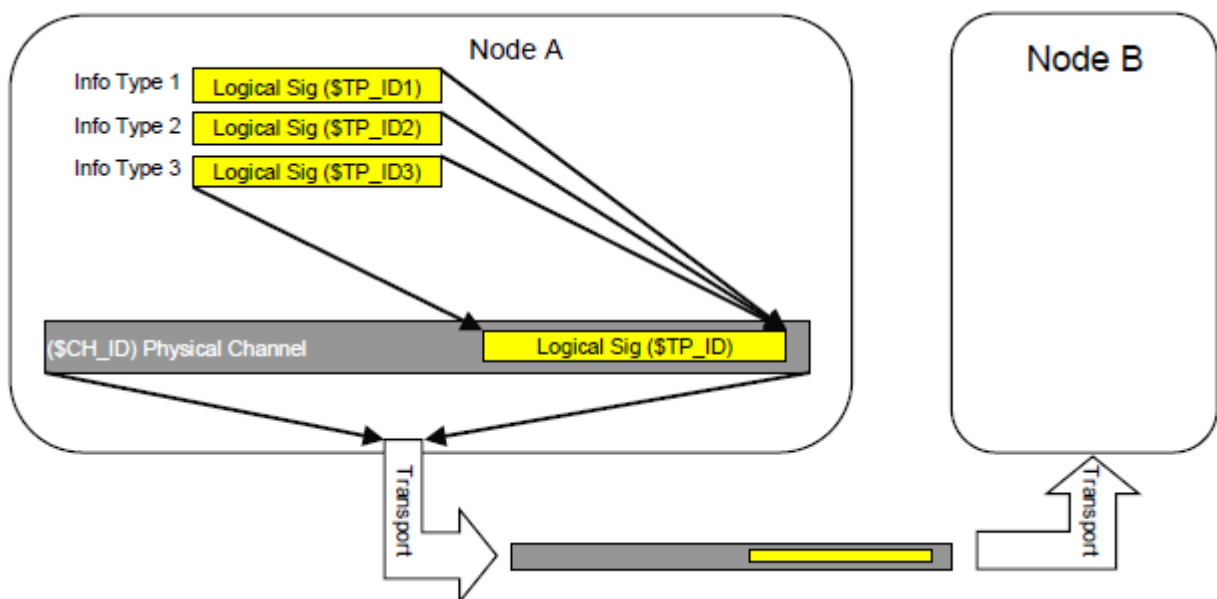
Table 9 — Summary of N_PCI bytes

N_PDU name	N_PCI bytes						
	Byte #1		Byte #2	Byte #3	Byte #4	Byte #5	Byte #6
	Bits 7 - 4	Bits 3 - 0					
SingleFrame (SF) (CAN_DL ≤ 8)	0000 ₂	SF_DL	—	—	—	—	—
SingleFrame (SF) (CAN_DL > 8) ^a	0000 ₂	0000 ₂	SF_DL	—	—	—	—
FirstFrame (FF) (FF_DL ≤ 4 095)	0001 ₂	FF_DL		—	—	—	—
FirstFrame (FF) (FF_DL > 4 095) ^b	0001 ₂	0000 ₂	0000 0000 ₂	FF_DL			
ConsecutiveFrame (CF)	0010 ₂	SN	—	—	—	—	—
FlowControl (FC)	0011 ₂	FS	BS	ST _{min}	N/A	N/A	N/A

下图就是一组中控实际发出的 TP 连续帧， MessageID=0x2B7。

131.07...	CAN 3	195	LBClient1_Request_Signals	CAN Frame	Rx	8	8	00 00 00 00 43 02 00 01
131.26...	CAN 3	2B7	CONMP_MC_WORD_Tx	CAN Frame	Rx	8	8	10 56 76 74 01 00 01 02
131.27...	CAN 3	2B7	CONMP_MC_WORD_Tx	CAN Frame	Rx	8	8	21 00 00 00 00 04 00 03
131.28...	CAN 3	2B7	CONMP_MC_WORD_Tx	CAN Frame	Rx	8	8	22 00 00 02 00 00 41 00
131.29...	CAN 3	2B7	CONMP_MC_WORD_Tx	CAN Frame	Rx	8	8	23 75 00 64 00 69 00 6F
131.30...	CAN 3	2B7	CONMP_MC_WORD_Tx	CAN Frame	Rx	8	8	24 00 00 00 01 20 11 07
131.31...	CAN 3	2B7	CONMP_MC_WORD_Tx	CAN Frame	Rx	8	8	25 96 8F 5F C3 54 2C 00
131.32...	CAN 3	2B7	CONMP_MC_WORD_Tx	CAN Frame	Rx	8	8	26 00 00 02 20 11 FF 00
131.33...	CAN 3	2B7	CONMP_MC_WORD_Tx	CAN Frame	Rx	8	8	27 55 00 53 00 42 00 3A
131.34...	CAN 3	2B7	CONMP_MC_WORD_Tx	CAN Frame	Rx	8	8	28 00 42 00 4F 00 4F 00
131.35...	CAN 3	2B7	CONMP_MC_WORD_Tx	CAN Frame	Rx	8	8	29 54 00 4B 00 45 00 59
131.36...	CAN 3	2B7	CONMP_MC_WORD_Tx	CAN Frame	Rx	8	8	2A 00 00 00 03 20 11 0A
131.37...	CAN 3	2B7	CONMP_MC_WORD_Tx	CAN Frame	Rx	8	8	2B 84 DD 72 59 97 F3 4E
131.38...	CAN 3	2B7	CONMP_MC_WORD_Tx	CAN Frame	Rx	8	8	2C 50 00 00 00 00 00 00
131.38...	CAN 3	195	LBClient1_Request_Signals	CAN Frame	Rx	8	8	00 00 00 00 00 00 00 00

因为多屏互动功能需要传输的信息较多，CAN 总线上实际可用的物理 CAN Message ID 数量有限，所以系统通过在标准的连续帧协议基础上，对上述协议中实际的数据区域做协议在封装，从而达到在单条物理 CAN Message 中复用多条逻辑信号的目的。

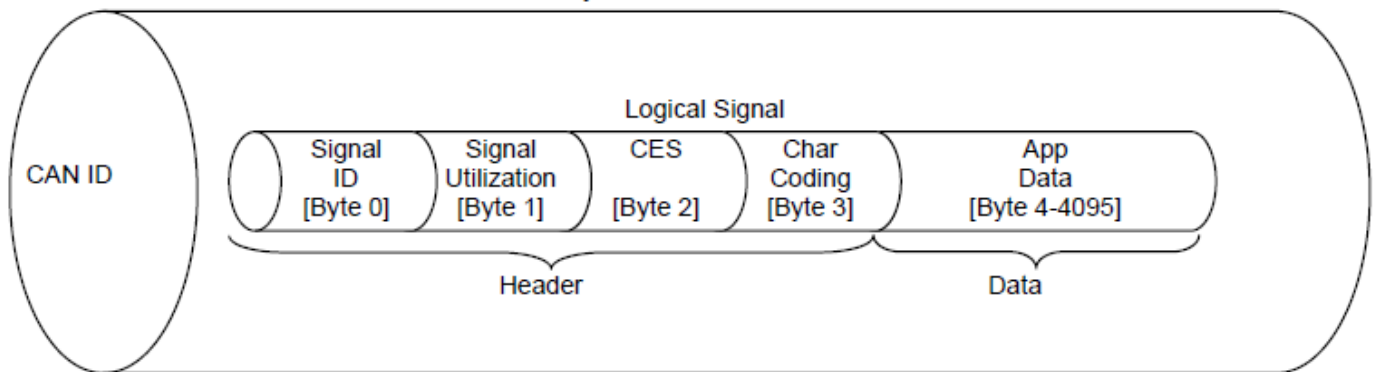


下图展示了福特的 TP 协议的具体帧结构，被填充在标准连续帧报文的数据区的逻辑信号会包含以下几个字段信息：

- Signal ID: 具体的逻辑信号 ID，按协议定义每条物理消息可以派生 255 条逻辑信号
- Signal Utilization: 逻辑信号用途属性，具体属性会在每条逻辑信号的数据结构中详细说明

- CES: Command Execution Status 指令执行状态, 用于回复对手件当前指令执行状态
- Char Coding: 编码方式, 用于定义后续应用数据区字符串的编码方式
- APP Data: 应用数据区, 存放具体的报文信息, 长度可变。

Physical Channel



- Signal Utilization 逻辑信号用途属性具体在 TP 协议的 **TP-REQ-015128/G-Signal Utilization (TcSE ROIN-138092-7)** 章节会定义各个属性的 ID, 在规范的后半部分具体定义每条逻辑信号的数据结构时候也会对每条逻辑信号的用途属性做定义, 一般一条逻辑信号只有一个固定的 Signal Utilization 属性。
- CES 指令执行状态会在 TP 协议的 **1.2.9 Command Execution Status** 章节详细定义各个状态需要回复的 CES 码, 这里需要注意中控在实际回复对手件 TP 报文时需要根据自身系统状态严格按表格要求回复 CES 码, 否则可能造成对手件响应超时不显示信息或对手件长期等待响应造成假死等问题。
- Char Code 编码方式主要有三种: 00 代表 UTF-16 编码, 此种编码方式可以支持中文等其他语言字符, 每个字符长度为 2Byte, 01 代表 Latin 编码, 此种编码方式只支持数字、符号和英文字符, 每个字符长度为 1Byte, 02 代表 RAW 原始数据, 此种编码方式下直接将原始数据通过 16 进制方式编码传输。
- EOS: End Of String 字符串截止符, 在数据区通常会传输多个字符串, 每个字符串之间使用 EOS 截止符来间隔开, 协议规定编码方式为 UTF-16 时 EOS 截止符长度为 2Byte 既 0x0000, 其他编码方式下 EOS 截止符长度为 1Byte 既 0x00

TP 协议规范从 **1.4 Signal and Channel Catalog** 章节开始大篇幅的用于定义每条具逻辑信号的具体数据结构, 以下将以多媒体娱乐信息 0x2B7 消息中的 0x79 信号为例, 来解读 SPEC 规范

首先在章节 **1.4.1.12 TP-PHY-TPP-REQ-023125/E-MediaPlayerServer - MediaPlayerClient (TcSE ROIN-160781-2)** 会定义 0x2B7 这条实际存在的物理 CAN 消息其可以传输的逻辑信号有哪些, 黄色高亮标出的就是需要进一步解析的逻辑信号 0x79, 信号名是多媒体信息, 用途属性是基本元数据。

			Channel		
CAN ID	Msg Name	TP Index	Transmitter: See CAN database		
0x2B7	CONMP_MC_WORD_Tx	18	Receiver: See CAN database		
			Logical Signals		
			Signal ID	Signal Name	Utilization
			0x79	MediaInformation_St	Generic Metadata
			0x76	LBP1_ItemInfo_Rsp	List Browser Data
			0x5F	GetTUPresetInfo_Rsp	AMFM, SDARS, DAB
			0x67	DispInfo_ArtistName_St	SDARS
			0x68	SDARS_CatName_St	SDARS
			0x6C	SDARS_ChannelName_St	SDARS
			0x6F	DispInfo_SongTitle_St	SDARS
			0xA9	ProjMdeMediaPlayerRepeater_St	Projection_Mode
			0xB3	MediaInformation2_St	Generic Metadata

接着在章节 **1.4.2.39 TP-LOG-TPL-REQ-023174/A-SID-79-MediaInformation_St (TcSE ROIN-160692-2)** 具体定义了 0x79 信号的数据结构。0x79 信号的前 4Byte 遵循协议的定义分别是 ID 为 0x79, Utilization 为 0x17 常规多媒体元数据, CES 和编码方式, 其中 CES 位根据指令请求时系统实际状态回复, 编码方式需要根据后续元数据内容决定, 如

果是纯英文的信息则可选择 Latin，如果歌曲信息为中文则必须选择 UTF-16.

Byte 0: Signal identifier

0x79: MediaInformation_St

Byte 1: Utilization

0x17: MP_Media7 – Generic Metadata

Byte 2: Command Execution Status

0x0y: Final Result – Success
0x1y: Final Result – Fail
0x2y: Final Result – Information
0x3y: Intermediate Result– Wait

Byte 3: Character Coding

Bit 0-5: Reserved

Bit 6-7: Coding

0x0: Coding Table I
0x0000-0xFFFF UNICODE UTF-16 (2 byte per char)
0x1: Coding Table II
0x00-0xFF Latin-9 (1 byte per char)

信号的第 4Byte 开始为实际的数据区，0x79 信号的数据区分别填充了数据状态（当前数据是否可用，当前音源是否有元数据信息），元数据 1 的图标，元数据 2 的图标，元数据 1，元数据 2 和数据源名称。
通常会将元数据 1 会被定义为歌曲名称，元数据 2 会被定义为艺术家名称，数据源名称会被填充当前的设备名称或者 APP 应用名称。
对手件会根据元数据 1 图标和元数据 2 图标这两个字段来判断后续元数据的属性，从而将不同的字符串解析后放置到屏幕的对应位置，具体的元数据图标 ID 号可以在 LBP 协议中找到。

Byte 4 up to 126/66 (Coding Table I / Coding Table II):										0x19 - 0xFF Reserved									
Byte 1:										Byte 3: Metadatacon_2									
Bits 0-2: Reserved										0x00 Invalid									
Bits 3 - 5: DataUpdate										0x01.. 0x18 IconID's									
0x0 Inactive										0x19 - 0xFF Reserved									
0x1 Set Operation										Byte 4:									
0x2 Data refresh										Metadata1									
Bits 6 - 7: NonMetadataSrc										Metadata1									
0x0 No										19 Characters Max plus 1 end of string character									
0x1 Yes										Metadata2									
Byte 2: Metadatacon_1										Metadata2									
0x00 Invalid										19 Characters Max plus 1 end of string character									
0x01.. 0x18 IconID's										SourceInformation									
										SourceInformation									
										19 Characters Max plus 1 end of string character									

以下为一组实际发送的 79 消息解析的结果

								Signal ID	Utilization	Exc. Sls	Coding	DataUpdate	NonMeta	MetaData	MetaData	Metadata1								Metadata2								Source Name							
10	1F	79	17	1	0	8	1	79	17	01	00	08	01	02	59	27	98	CE	54	39	00	00	73	8B	8D	6B	91	CE	00	00	96	8F	5F	C3	54	2C	00	00	
21	2	59	27	98	CE	54	39	Media Info	Gen Metadata	Information	UTF-16	Inactive Yes	Song Title	Music Artist																									
22	0	0	73	8B	8D	6B	91																																
23	CE	0	0	96	8F	5F	C3																																
24	54	2C	0	0	CC	CC	CC																																

2. LBP 协议

对于需要做列表显示和控制的信息需要用到 LBP 协议，LBP 协议中列表结构如下。

Item Index	Data Type	Activation Event	Object Type	Object State	Item Descriptor
0x0001	0x00-0xFF	Supported/Not Supported	Entry/List	Active/Inactive	{Descriptor Tag}
..
0xFFFF

完整的列表由以下几部分组成：

- 条目序号：列表中每个条目会对应一个唯一的序号
- 数据类型：数据类型编码会定义后续 Item Descriptor 里所包含的信息，详细定义可以在 LBP 协议规范的 1.4.6

LBPv2-SR-REQ-128955/G-Item Descriptor Attribute 章节之后的表格中找到对应关系。下图为表格部分截图，黄色高亮的代表的数据类型是媒体源类型，其条目描述里包含了媒体源图标和媒体源名称两个字段。

Encoding	Data Type	Item Descriptor	Descriptor Length
0x00	General Reserved		
0x01	General Reserved		
0x02	Generic Text	{Generic Text}	25 characters + EOS
...	General Reserved		
0x1F	General Reserved		
0x20	Media Type	{SourceIcon}{MediaSourceName}	{Source Icon, \$0 = Invalid } {25 characters + EOS}

- Activation Event: 代表的是当前这个条目在列表中是否可以被选中，通常列表的表头该属性为 Not Supported。
- Object Type: 代表该条目是作为二级菜单的入口，还是作为列表中一个可以被执行的项。前者用户选中后对手件会继续请求下一级菜单，此时中控需要准备并回复下一级菜单的列表内容。后者用户选中后对手件会直接发起一个执行请求，中控收到该请求后需要执行约定好的动作例如切换到对应的音源。
- Object Status: 代表当前条目是否处于激活状态，如果是未激活状态则对手件不会显示对应内容。
- Item Description: 代表该条目具体在对手件上需要显示的信息，显示内容由上述 Object Type 决定。

以下为一个完整的多媒体音源列表

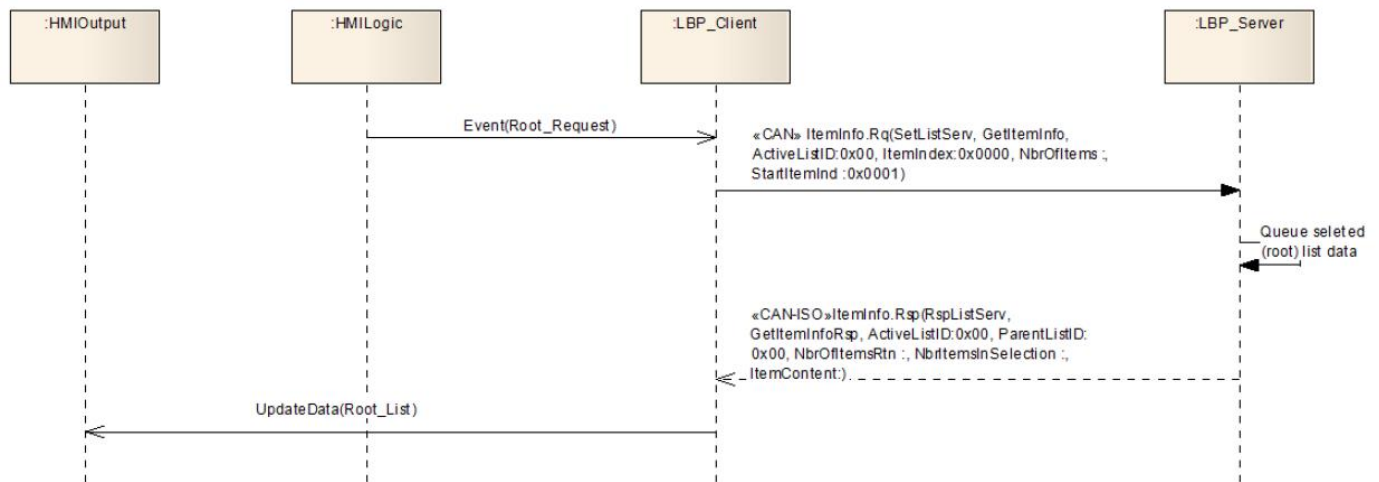
Media Root List						
ListServerID =	Generic Media1 (0x02)					
ActiveListID =	Media Root (0x0000)					
ParentList ID =	Media Root (0x0000)					
NbrItemsInSelection=	20					
ItemIndex	Data Type	ActivationEvent	Object Type	ItemDescriptor	getitem(ItemIndex) Behavior	setItem() Behavior
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label	{Media Root List}	Invalid	Invalid
0x0001	Media Type (0x20)	Supported (0x1)	List Object (0x2)	{Source Icon} {Source # Source Name}	Goto Active List ID 0x0001	Set Source to Entry 1
0x0002	Media Type (0x20)	Supported (0x1)	List Object (0x2)	{Source Icon} {Source # Source Name}	Goto Active List ID 0x0001	Set Source to Entry 2
0x0003	Media Type (0x20)	Supported (0x1)	List Object (0x2)	{Source Icon} {Source # Source Name}	Goto Active List ID 0x0001	Set Source to Entry 3
0x0004	Media Type (0x20)	Supported (0x1)	List Object (0x2)	{Source Icon} {Source # Source Name}	Goto Active List ID 0x0001	Set Source to Entry 4
...						
0x0014	Media Type (0x20)	Supported (0x1)	Entry Object (0x1)	{Source Icon} {Source # Source Name}	Invalid	Set Source to Entry 20 Non Indexing Device can only use setItem
Example 1:						
0x0001	Media Type (0x20)	Supported (0x1)	List Object (0x2)	{USB Icon} 1:USB	Goto Active List ID 0x0001	Set Source to Entry 1
0x0002	Media Type (0x20)	Supported (0x1)	List Object (0x2)	{USB Icon} 2:USB	Goto Active List ID 0x0001	Set Source to Entry 2
Example 2:						
0x0001	Media Type (0x20)	Supported (0x1)	Entry Object (0x1)	{0x07} 1: Pandora	Invalid	Set Source to Entry 1
0x0002	Media Type (0x20)	Supported (0x1)	Entry Object (0x1)	{0x07} 2: Heart Radio	Invalid	Set Source to Entry 2
0x0003	Media Type (0x20)	Supported (0x1)	Entry Object (0x1)	{0x07} 3: CarPlay	Invalid	Set Source to Entry 3
0x0004	Media Type (0x20)	Supported (0x1)	Entry Object (0x1)	{0x0A} 4: BTAudio	Invalid	Set Source to Entry 4
Example: Now Playing (Applink Source selected)						
Source Info: (device name)						
Metadata1: (mainField 1)						
Metadata2: (mainField 2)						
Metadata Icon 1 and 2: 0x17 (No Icon)						

Note: Fields described in examples are referenced in S28 63.4.2.41 TP-LOG-GTPC-16082-2-SID-79-MediaInformation_St

Indexing Device Features						
ListServerID =	Generic Media1 (0x02)					
ActiveListID =	Indexable Features (0x0001)					
ParentList ID =	Media Root (0x0000)					
NbrItemsInSelection=	4					
ItemIndex	Data Type	ActivationEvent	Object Type	ItemDescriptor	getitem(ItemIndex) Behavior	setItem() Behavior
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label	{Device Name}	Invalid	Invalid
0x0001	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	Now Playing	Invalid	Play Entry 1
0x0002	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	Play All	Invalid	Play Entry 2
0x0003	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	Shuffle All	Invalid	Play Entry 3
Example:						
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label	Ken's iPod	Invalid	Invalid

列表信息的请求和回复时序如下图，客户端会根据用户的实际操作发起一个列表请求动作，对应的 CAN 消息会包含 SetListServ (列表服务号，用于确定请求的是什么类型的列表)，GetItemInfo (操作状态为获取条目信息)，ActiveListID (获取对应列表的 ID)，ItemIndex (条目编号)，NbrOfItems (获取列表条目数量)，StartItemInd (从列表的第几个条目开始获取)。中控收到这个请求后需要在本地准备好队列，完成准备后再通过 TP 报文回复客户端对应的列表信息，回复信息中新增了两个字段分别为 NbrOfItemsRtn (当前回复的条目数) 和 NbrItemInSelection (列

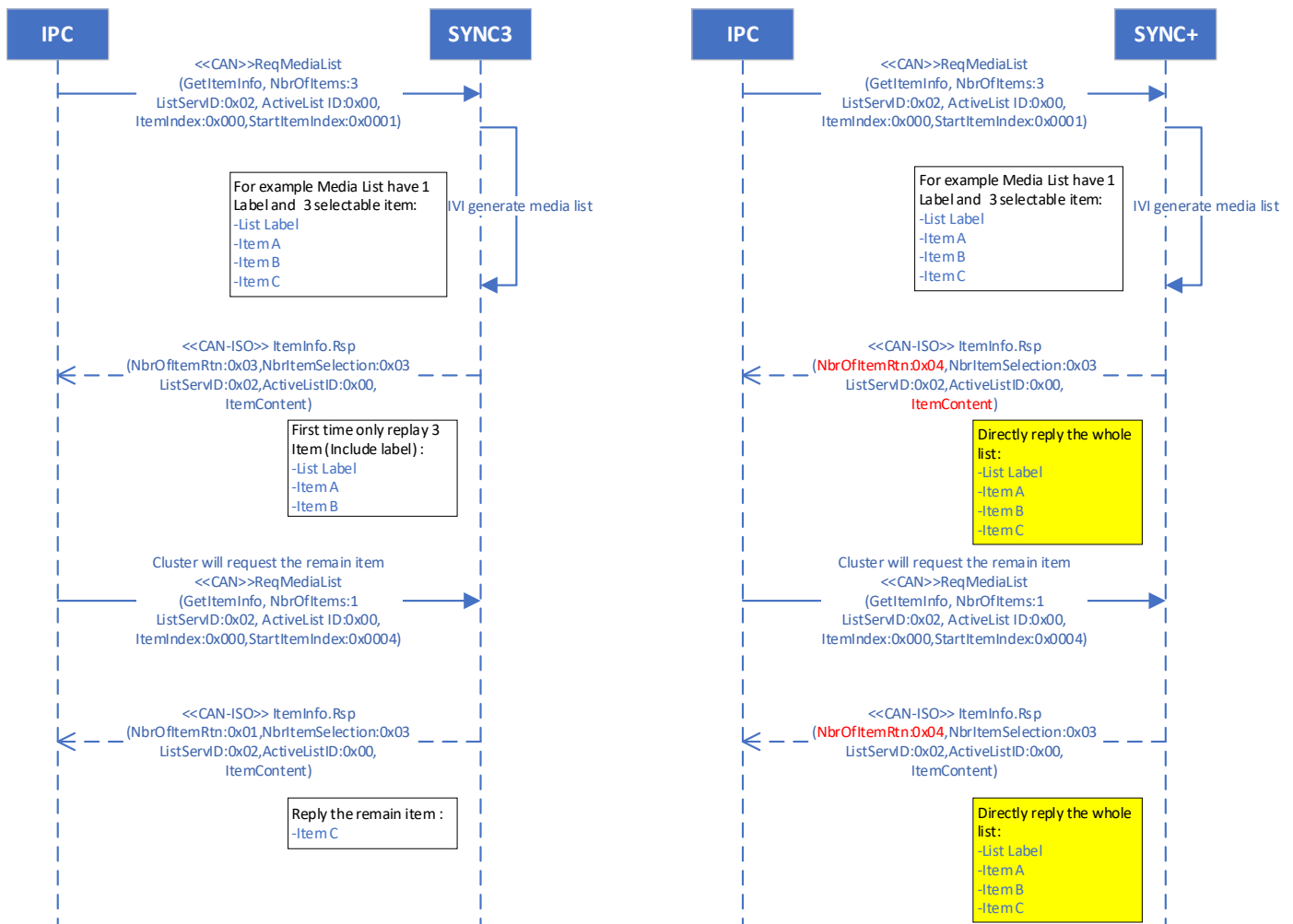
表中总的可以选条目数)。



这里需要注意回复列表时，回复的条目数必须与请求的数量匹配，否则部分对手件会判定为非法回复而抛弃接收到的内容。例如中控侧存在一个长度为 4 的列表（1 个表头+3 个可选项），会出现以下三种请求：

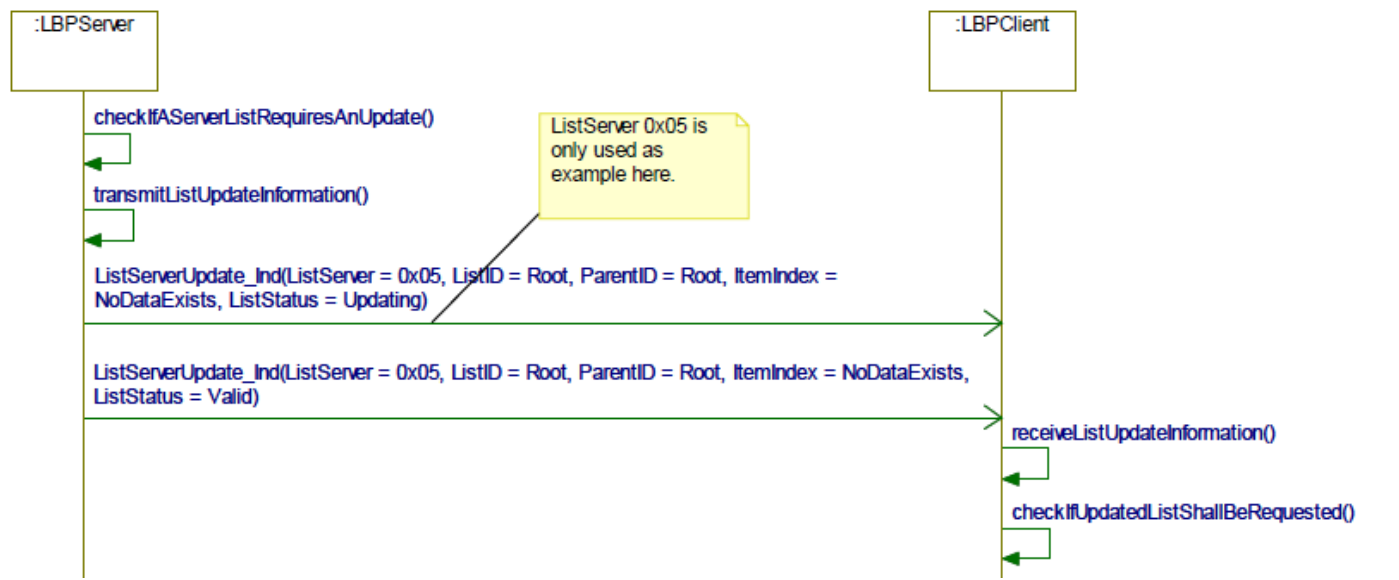
1. 对手件请求获取列表长度为 4，正好与中控侧列表长度匹配，中控直接回复即可。
2. 对手件请求获取列表长度为 5，大于中控侧列表长度，此时中控按实际情况回复列表长度为 4 即可
3. 对手件请求获取列表长度为 2，小于中控侧列表长度，此时中控只能回复前两条条目并且将 NbrOfItemsRtn 设成 2，但是 NbrItemInSelection 需按实际情况设置成 3。对手件在收到报文后会根据返回的 NbrItemInSelection 来判断实际列表总长度为 4 而自己只请求了两条，此时对手件通常会有两种做法，第一种直接重新请求一个长度为 4 的列表，获取完整的列表信息，第二种是再请求一个长度为 2 的列表但是起始位从 3 开始，来获取剩下的两条信息。

下图左侧序列图展示了正确的列表信息回复方式，左侧展示错误的回复方式



实际操作过程中一些对手件不会频繁的请求列表信息（列表信息需要使用大量的连续帧 CAN 消息来传输，会加大 CAN 总线负载），那么一旦中控侧的列表信息发生变化，则会出现 LBP 协议两端所持有的列表信息不匹配问题。为了解决该问题，又避免频繁查询对 CAN 总线造成较高负载，规范引入了列表更新通知机制，具体在 LBP 协议的 2.11.2.1.1 LBP-SD-REQ-130402/A-Indicate server list update 章节定义。

时序图如下，当主机端判断到有列表更新时需要发起一个 ListServerUpdate 的事件消息给客户端来通知，消息会包含列表的服务 ID、列表 ID、父列表 ID，需要注意的是此时主机端必须按时序图要求连发两帧消息，第一帧列表状态为 Updating，第二帧状态为 Valid，客户端侧只有看到 Updating → Valid 的上升沿后才会发起重新请求列表的动作。



3. 应用要点

以上章节主要介绍了基本 TP 和 LBP 协议栈的数据结构定义和交互方式。具体个应用层的信息填充规则和填充的时序在 NavRepeater、蓝牙电话、多媒体的相关应用 Spec 中都会有详细定义。以下会对之前项目应用过程中的一些数据使用注意点做总结。

➤ 3.1 导航图标信息

在 NavRepeater 规范中定义了一个导航诱导图标是通过前方路口分叉数+各个分叉属性数组（朝向+属性）的方式来定义的，简单点说一个诱导图标的代码=N+O1+T1+...+On+Tn（O 代表朝向，T 代表属性）来表示的，具体可以参考 NavRepeater 文档定义，理论上通过以上方式可以组合出上万种路口图标。但实际操作过程中由于 IPC 和 HUD 内存限制，只会预存常用的图标，为了方便供应商开发，以下表格直接将仪表中已存图标和百度地图所用的图标信息作了映射，软件开发人员只需要按以下表格查表返回对应代码即可，不需要理解图标组包规则。

编号	回调事件	百度地图效果	福特仪表图标					备注
1	"turn_back.png", /*掉头 */			01	FF	04		
2	"turn_front.png", /*直行 */			01	00	11		
3	"turn_right_front.png", /*右前方转弯 */			01	F0	03		

4	"turn_right.png", /*右转 */			01	00	03			
5	"turn_right_back.png", /*右后方转弯 */			01	A0	03			
6	"turn_right_side.png", /*普通/JCT/SAPA 二分歧靠右 */			02	00	02	C0	09	
7	"turn_right_side_ic.png", /**< IC 二分歧右侧走 IC */			02	00	01	E0	0A	
8	"turn_left_back.png", /* 左后方转弯 */			01	60	03			
9	"turn_left.png", /*左转 */			01	40	03			
10	"turn_left_front.png", /*左前方转弯 */			01	20	03			
11	"turn_left_side.png", /* 普通/JCT/SAPA 二分歧靠左 */			02	00	02	40	09	
12	"turn_left_side_main.png", /* 左侧走本线 */			01	00	11			
13	"turn_left_side_ic.png", /*二分歧左侧走*/			02	00	01	20	0A	
14	"turn_start.png", /* 起始地 */		N.A.					仪表是否不需要	
15	"turn_dest.png", /*目的地 */			01	C0	13			
16	"turn_via_1.png", /*途经点 1 */			02	01	30	C0	14	
17	"turn_inferry.png", /* 进入渡口 */		N.A.					建议可用小房子或者 其他特殊 icon 替代	
18	"turn_tollgate.png", /*收费站 */		N.A.						
19	"turn_right_side_main.png", /*普通/JCT/SAPA 二分歧右侧直行*/			01	00	11			
20	"turn_branch_left_straight.png", /*普通/JCT/SAPA 三分歧左侧直行*/			01	00	11			
21	"turn_branch_center.png", /*普通/JCT/SAPA 三分歧中央直行*/			01	00	11			

22	"turn_branch_right_straight.png", /*普通/JCT/SAPA 三分歧右侧直行*/			01	00	11			
23	"turn_branch_left.png", /*三分歧左侧走*/			01	20	03			
24	"turn_branch_right.png", /*三分歧右侧走 */			01	F0	03			
25	"turn_lf_2branch_left.png", /* 八方向左前方靠左侧 */			01	20	03			建议用左转替代
26	"turn_lf_2branch_right.png", /*八方向左前方靠右侧 */			01	F0	03			
27	"turn_rf_2branch_left.png", /*八方向右前方靠左侧 */			01	20	03			建议用右转替代
28	"turn_rf_2branch_right.png", /*八方向右前方靠右侧 */			01	F0	03			
29	"turn_ring_front.png", /*环岛向前 */			01	00	0C			
30	"turn_ring_left.png", /* 环岛向左 */			01	40	0C			
31	"turn_ring_leftback.png", /* 环岛向左后 */			01	60	0C			
32	"turn_ring_leftfront.png", /*环岛向左前 */			01	20	0C			
33	"turn_ring_right.png", /*环岛向右 */			01	C0	0C			
34	"turn_ring_rightback.png", /*环岛向右后 */			01	A0	0C			
35	"turn_ring_rightfront.png", /*环岛向右前 */			01	E0	0C			
36	"turn_ring_back.png", /*环岛向后 */			01	FF	04			用环岛向前替代?
37	"turn_ring.png", /* 环岛 */			01	00	0C			
38	"turn_ring_out.png", /* 环岛出口 */			01	00	0C			

➤ 3.2 如何解决切换下一首歌曲后仪表还显示上一曲信息的延时问题

从系统设计上来看歌曲信息是通过 TP 消息传输，由于 TP 消息是多帧传输的，所以整包完成传输会需要一定时间，这就造成了信息传输的延时（通常在 100~200ms 左右），最终产生的效果是 IVI 侧已经完成了切歌的动作并开始播放，但是仪表侧还会停留在上一曲歌名，然后再刷新到新的歌名。为了解决此问题，服务层可以做以下设计，但执行切歌动作是先发送一组空的歌曲名称 TP 报文，由于空名称很短所以对应 TP 报文只需要一个单帧即可传输完成，仪表侧接收到空报文会将上一曲的歌名等信息用空替换，然后 IVI 再正常发送歌名，此时在仪表侧看到的完整动作是切歌时上一曲信息会先消失，下一曲信息会延时个 200ms 完成显示。

➤ 3.3 如何解决来电时候仪表先显示上一次通话信息问题

与上一个问题一样，来电状态是通过标准 CAN 消息传输没有延时，而来电人姓名电话等信息是通过 TP 消息传输的有传输延时问题，当来电时来电状态第一时间传输到了仪表，仪表通过判断来电状态判定需要显示来电人姓名，但此时来电人姓名还在通过多帧传输协议传输，仪表本地缓存中上一次来电人姓名还没有被本次有效的信息刷新，所以会造成仪表上先显示了上一次来电人姓名，当本次来电人姓名的多帧消息完成传输仪表本地缓存被刷新后，仪表才会在显示本次来电人姓名。连贯起来看就是来电时仪表会先闪现上一次来电人姓名，再显示当前来电人姓名。解决此问题的方法同样是当电话挂断时，多屏互动服务层需要补发一次空的来电人姓名给仪表，用来擦除仪表本地缓存，这样当下一次来电时仪表可以先显示空的信息。