

فهرست مطالب

8.....	1- دیاگ چیست
13.....	1-1- کاربرد پایه های OBDII
13.....	1-2- محل نصب پورت OBDII
14.....	1-3- پروتکل های متداول در OBDII
16.....	1-4- کاربردهای OBD
16.....	1-5- معرفی کدهای خطا
17.....	1-5-1- دسته بندی کد خطا
18.....	2- معرفی پروتکل KLine
19.....	2-1- بسته استاندارد داده در Kline
19.....	2-2- سخت افزار لازم برای اتصال به میکروکنترلر
21.....	3- قالب بندی داده ها در K-Line
21.....	3-1- قالب بندی پیام درخواست
22.....	3-1-1- بخش هدر
23.....	3-1-2- بایت آدرس هدف
23.....	3-1-3- بایت آدرس منبع
24.....	3-1-4- بایت طول پیام
25.....	3-1-5- بایت های داده
26.....	3-1-6- بایت چک سام
26.....	4- قالب بندی پیام درخواست
28.....	5- زمان بندی در ارتباط KLine
30.....	5-1- مستثنات زمان بندی

- 6- تبادل داده در KLine..... 30
- 6-1 رویه شروع ارتباط..... 31
- 6-2 ایجاد ارتباط سریع..... 32
- 7- شروع کند..... 35
- 8- انواع درخواست‌ها به ECU..... 36
- 8-1 الگوریتم پاسخ‌دهی ECU به درخواست ارسال شده به آن..... 37
- 8-2 درخواست شروع ارتباط..... 40
- 8-3 درخواست پایان ارتباط..... 40
- 8-4 درخواست دسترسی به پارامترهای زمان‌بندی..... 41
- 8-5 درخواست اعلان حضور..... 42
- 8-6 درخواست ریست واحد الکترونیکی..... 43
- 8-7 درخواست خواندن اطلاعات واحد الکترونیکی..... 44
- 8-8 سرویس درخواست شروع نشست..... 45
- 8-9 درخواست پایان نشست..... 47
- 8-10 درخواست خواندن خطا..... 49
- 8-10-1 کد خطا 18..... 49
- 8-10-2 کد خطا 13..... 51
- 8-10-3 کد خطا 17..... 52
- 8-11 درخواست پاک کردن خطا..... 53
- 8-12 سرویس خواندن داده..... 54
- 8-12-1 سرویس خواندن داده با شناسه محلی..... 54
- 8-12-2 سرویس خواندن داده با شناسه عمومی..... 55
- 8-13 سرویس کنترل ورودی/خروجی (تست عملگرها)..... 55

- 8-13-1- سرویس کنترل ورودی/خروجی با شناسه محلی 55
- 8-13-2- سرویس کنترل ورودی/خروجی با شناسه عمومی 57
- 8-13-3- رویه تست عملگرها 58
- 9- هندل کردن خطا 58
- 9-1- هندل کردن خطا در هنگام شروع ارتباط 58
- 9-2- هندل کردن خطا در هنگام تبادل داده 58
- 9-2-1- شناسایی خطا توسط ECU در هنگام ارسال داده 58
- 9-2-2- شناسایی خطا توسط درخواست کننده در پیام پاسخ 58
- 9-2-3- ECU تشخیص می دهد که در پاسخی که ارسال کرده خطا دارد 58
- 9-2-4- درخواست کننده تشخیص دهد که در پیام درخواست خطا وجود دارد 59

فهرست شکل‌ها

- شکل 1- تاریخچه OBD 9
- شکل 2- اولین برد پورت OBD 10
- شکل 3- پورت ALDL شرکت جنرال موتور 10
- شکل 4- استاندارد پورت دیاگ در طول زمان 11
- شکل 5 - پورت OBDII 12
- شکل 6 - محل نصب پورت OBDII در خودرو 14
- شکل 7- تاریخچه استفاده از پروتکل‌ها در OBDII 14
- شکل 8 - شرح کد خطا 17
- شکل 9- بسته استاندارد داده در KLine 19
- شکل 10- مدار مبدل KLine به UART 20
- شکل 11- مدار اتصال پین LLine و پین کنترل کننده KLine 21
- شکل 12 - ساختار پیام در KLine 21
- شکل 13- ساختار بایت قالب بندی در KLine 22
- شکل 14- آدرس ECU و درخواست کننده برای KWP 23
- شکل 15 - ساختار پیام در KLine بدون بایت های آدرس و طول 25
- شکل 16 - ساختار پیام در KLine با بایت های آدرس ولی بدون بایت طول 25
- شکل 17 - ساختار پیام در KLine بدون بایت های آدرس و با بایت طول 25
- شکل 18 - ساختار پیام در KLine با بایت های آدرس و بایت طول 25
- شکل 19- ساختار پیام پاسخ مثبت ECU به یک درخواست در ارتباط KLine 26
- شکل 20- ساختار پیام پاسخ منفی ECU به یک درخواست در ارتباط KLine 27
- شکل 21 - زمان بندی و جریان داده در پروتکل KLine 29
- شکل 22- رویه شروع درخواست در KLine 31
- شکل 23- الگوریتم ایجاد ارتباط در KLine 32
- شکل 24- زمان بندی شروع سریع در KLine 33
- شکل 25- ساختار پیام درخواست شروع ارتباط در KLine 33
- شکل 26- ساختار پیام جواب مثبت در KLine 34
- شکل 27- ساختار بایت KB1 در KLine 34
- شکل 28- زمان بندی شروع کند در KLine 36
- شکل 29- الگوریتم پاسخ دهی به درخواست در KLine 40

- شکل 30 - ساختار پیام درخواست پایان ارتباط در KLine 40
- شکل 31 - ساختار پیام جواب مثبت به درخواست پایان ارتباط در KLine 40
- شکل 32 - ساختار پیام جواب منفی به درخواست پایان ارتباط در KLine 41
- شکل 33 - الف) ساختار پیام درخواست دسترسی به زمان بندی در KLine ب) فرمت ارسال پارمترهای تنظیم زمان بندی. 41
- شکل 34 - ساختار پیام جواب مثبت به درخواست دسترسی به زمان بندی در KLine 42
- شکل 35 - ساختار پیام جواب منفی به درخواست دسترسی به زمان بندی در KLine 42
- شکل 36 - ساختار پیام اعلان حضور در KLine 43
- شکل 37 - ساختار پیام جواب مثبت به درخواست اعلان حضور در KLine 43
- شکل 38 - ساختار پیام پاسخ منفی به درخواست اعلان حضور در KLine 43
- شکل 39 - ساختار پیام درخواست ریست در KLine 44
- شکل 40 - ساختار پیام پاسخ مثبت به درخواست ریست در KLine 44
- شکل 41 - ساختار پیام جواب منفی به درخواست ریست در KLine 44
- شکل 42 - ساختار پیام درخواست دسترسی به مشخصات در KLine 45
- شکل 43 - ساختار پیام جواب مثبت به درخواست دسترسی به مشخصات در KLine 45
- شکل 44 - ساختار پیام جواب منفی به درخواست دسترسی به مشخصات در KLine 45
- شکل 45 - نمودار حالت نشت‌های عیب‌یابی. 46
- شکل 46 - ساختار پیام درخواست شروع سشن در KLine 46
- شکل 47 - ساختار پیام جواب مثبت به درخواست شروع سشن در KLine 47
- شکل 48 - ساختار پیام جواب منفی به درخواست شروع سشن در KLine 47
- شکل 49 - ساختار پیام درخواست پایان سشن در KLine 48
- شکل 50 - ساختار پیام جواب مثبت به درخواست پایان سشن در KLine 48
- شکل 51 - ساختار پیام جواب منفی به درخواست پایان سشن در KLine 48
- شکل 52 - ساختار پیام درخواست خواندن خطاها در KLine 49
- شکل 53 - ساختار پیام جواب مثبت به درخواست خواندن خطا در KLine 50
- شکل 54 - روش ساخت کد خطا با توجه به داده ارسالی از واحد الکترونیکی. 50
- شکل 55 - ساختار پیام جواب منفی به سرویس درخواست خواندن خطا در KLine 50
- شکل 56 - ساختار پیام درخواست خواندن خطاها در KLine 51
- شکل 57 - ساختار پیام جواب مثبت به درخواست خواندن خطا در KLine 51
- شکل 58 - ساختار پیام جواب منفی به سرویس درخواست خواندن خطا در KLine 51
- شکل 59 - ساختار پیام درخواست خواندن خطاها در KLine 52

- شکل 60- ساختار پیام جواب مثبت به درخواست خواندن خطا در KLine 52
- شکل 61- ساختار پیام جواب منفی به سرویس درخواست خواندن خطا در KLine 52
- شکل 62- ساختار پیام درخواست پاک کردن خطاها در KLine 53
- شکل 63 - ساختار پیام جواب مثبت به درخواست پاک کردن خطاها در KLine 53
- شکل 64 - ساختار پیام جواب منفی به درخواست پاک کردن خطاها در KLine 53
- شکل 65- ساختار پیام خواندن داده با شناسه محلی حضور در KLine 54
- شکل 66 - ساختار پیام جواب مثبت به درخواست دسترسی به مشخصات در KLine 54
- شکل 67- ساختار پیام جواب منفی به درخواست خواندن داده در KLine 54
- شکل 68- ساختار پیام خواندن داده با شناسه محلی حضور در KLine 55
- شکل 69- ساختار پیام جواب مثبت به درخواست دسترسی به مشخصات در KLine 55
- شکل 70- ساختار پیام جواب منفی به درخواست خواندن داده در KLine 55
- شکل 71- ساختار پیام درخواست تست عملگر در KLine 56
- شکل 72- ساختار پیام جواب مثبت به درخواست تست عملگر در KLine 56
- شکل 73- ساختار پیام جواب منفی به درخواست تست عملگر در KLine 56

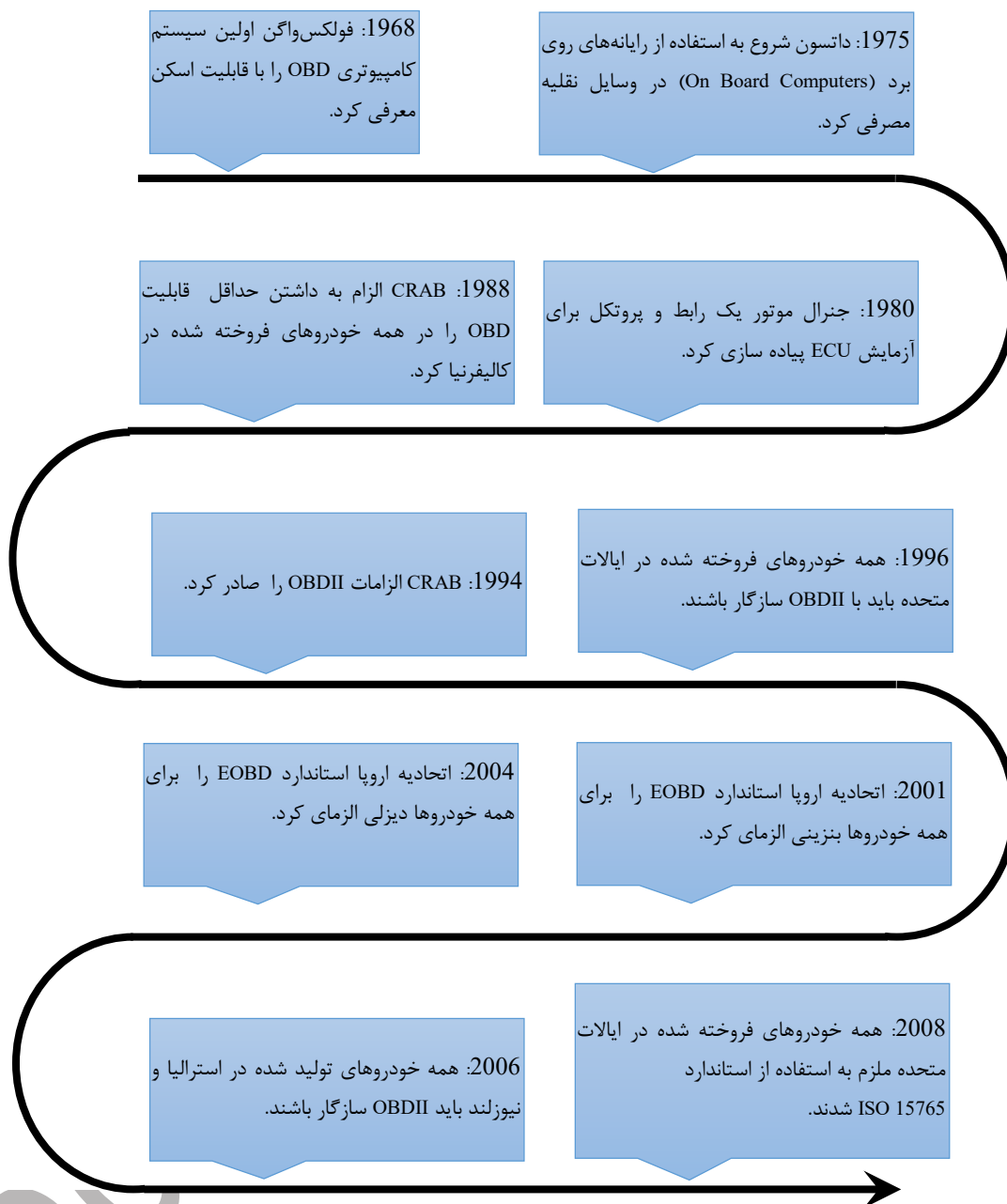
فهرست جدول‌ها

- جدول 1- توضیح پایه های OBDII 13
- جدول 2- انواع مد آدرس دهی در KLine 22
- جدول 3- چگونگی تعیین طول پیام در KLine 24
- جدول 4- انواع کدهای خطا در ارتباط KLine 27
- زمان بندی ارتباط KLine در حالت معمول به صورت شکل 9 است. توصیف زمان های P_1 - P_4 در جدول 5 آورده شده است.
- 28
- جدول 6- توصیف پارامترهای زمان بندی در KLine 29
- جدول 7- مقادیر متعارف پارامترهای زمان بندی برای آدرس دهی فیزیکی و فانکشنال. 29
- جدول 8- مقادیر متعارف پارامترهای زمان بندی برای آدرس دهی فیزیکی. 30
- جدول 9- مقادیر ممکن برای بایت KB1 در KLine 34
- جدول 10- حالات ممکن برای بایت KB1 35
- جدول 11- همه انواع درخواست ها در KLine 36

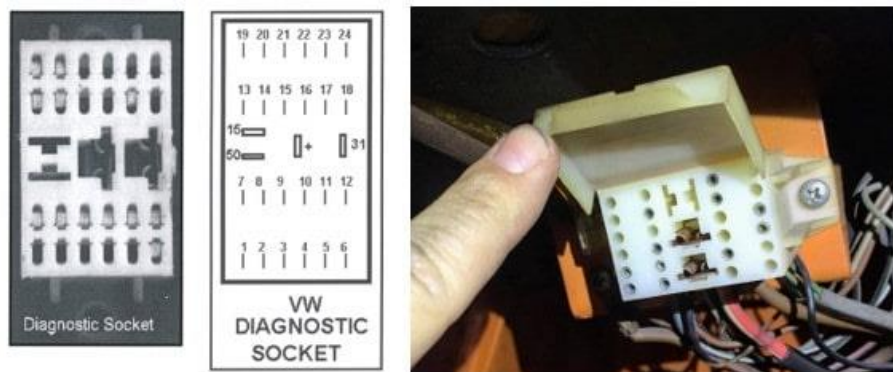
1- دیاگ چیست

OBD مخفف On-Board Diagnostic به معنای عیب‌یابی روی برد و به مفهوم سیستم استاندارد است که به دستگاه‌های الکترونیکی اجازه می‌دهد که با واحدهای الکترونیکی خودرو ارتباط برقرار نمایند. از طریق این ارتباط می‌توان مشکلات ایجاد شده در سیستم خودرو را بررسی کرد. از آنجایی که خودروها به طور فزاینده‌ای رایانه‌ای شده‌اند، این سیستم اهمیت بیشتری پیدا کرده است. در ابتدا این استاندارد برای کاهش انتشار گازهای گلخانه‌ای با نظارت بر عملکرد اجزای اصلی موتور توسعه داده شد. علاوه بر این، هدف از استفاده آن عیب‌یابی راحت‌تر سیستم تزریق سوخت الکترونیکی بود.

تاریخچه OBD در شکل 1 ترسیم شده است. اولین سیستم برای عیب‌یابی خودرو در سال 1968 توسط شرکت فولکس استفاده شد. با توجه به اینکه در آن زمان هنوز استاندارد در این زمینه تدوین نشده بود، پورت مخصوص به خود را طراحی و ارائه کرد (شکل 2).

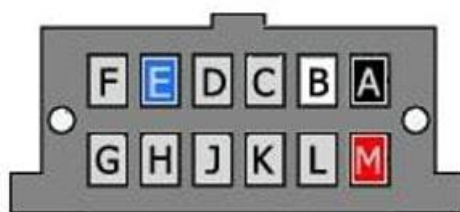


شکل 1- تاریخچه OBD.



شکل 2- اولین برد پورت OBD.

به این ترتیب نسخه اول دستگاه عیب‌یابی معرفی شد. اسم این پورت VW Diagnostic Socket بود. چند سال بعد در سال 1980 استاندارد ADLA¹ توسط شرکت جنرال موتور معرفی شد (شکل 3). بعدها این سوکت از اولین‌های OBDI شناخته شد. بر خلاف پورت شرکت فولکس واگن که فقط با ECU ارتباط داشت پورت شرکت جنرال موتور علاوه بر ECU با بقیه واحدهای الکترونیکی هم اتصال داشت.



شکل 3- پورت ALDL شرکت جنرال موتور.

بیشتر پروتکل‌ها با هدف کنترل آلاینده‌گی ارایه می‌شدند و بحث عیب‌یابی (به عنوان مثال تست عملگرها) به ندرت تعبیه می‌شد. داستان OBD از هنگامی شروع شد که هیئت منابع هوایی کالیفرنیا (CARB²) با هدف کنترل آلودگی هوا در سال 1988 الزام کرد که تمام اتومبیل‌های فروخته شده در کالیفرنیا به نوعی به قابلیت OBD مجهز شوند. در نتیجه استاندارد OBDI معرفی شد و استانداردسازی عیب‌یابی روی برد آغاز شد. اما تا سال ۱۹۹۱ اجباری به استانداردسازی وجود نداشت. استاندارد OBDI در ادامه توسعه پورت‌های قبلی و با همان اهداف به دنیا معرفی شد و به طور رسمی این استاندارد در سال 1991 اجرایی شد. اگرچه بسیاری از خودرو سازان این استاندارد را پذیرفتند اما هر شرکتی پورت مخصوص به خود را به همراه کابل واسط ارایه می‌کرد. این امر مشکلاتی

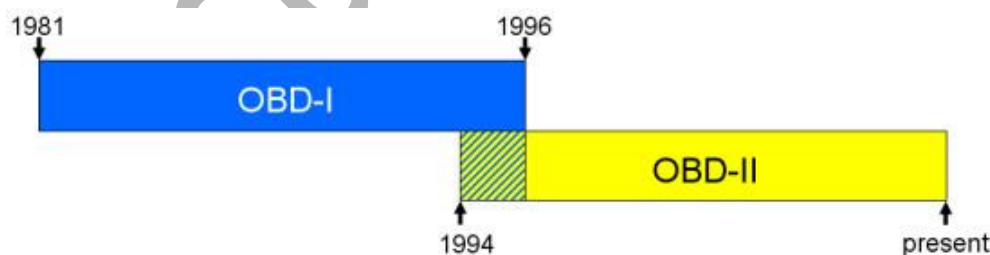
¹ Assembly Line Diagnostic Link

² California Air Resources Board

را برای تعمیرکاران ایجاد می‌کرد زیرا مکانیک باید برای هر خودرو کانکتور رابط مخصوص به خودش را استفاده می‌کرد. با این حال، CARB تا سال 1994 استانداردهای لازم را برای این سیستم‌ها صادر نکرد.

نوع	رابط استاندارد	رابط
1	2 پین (DB37)	
2	12 پین	
3	16 پین	
4	17 پین	

تفاوت در استانداردها، معرفی OBDII را ضروری کرد مشکلات ذکر شده شرکت‌های تولیدکننده را به این واداشت تا یک فرم کلی از پورت OBD ارائه دهند که در همه خودروها استفاده شود. در سال 1989 به صورت آزمایشگاهی و در سال 1996 به صورت رسمی رابط OBDII در خودروها عرضه شد (شکل 4). OBDII دارای پروتکل‌های سیگنالینگ پیشرفته است که مجموعه وسیعی از پارامترهای انتشار را می‌خواند. استاندارد OBDII کارایی بهتری در بررسی موتور و عملکرد آن انجام می‌دهد. در برخی موارد حتی به اصلاح آنها کمک می‌کند. علاوه بر این، OBDII کارایی موتور را بررسی می‌کند، به همین دلیل است که تا حد زیادی مترادف با چراغ چک موتور است. در مورد OBDI، حتی با پیشرفته‌ترین دستگاه‌های دیاگ، فقط می‌توانید کدهای خطا را بخوانید، عیوب را مشخص نمی‌کند و تست‌های بازده موتور مشابه OBDII را انجام نمی‌دهد.



شکل 4- استاندارد پورت دیاگ در طول زمان.



شکل 5 - پورت OBDII.

سوکت عیب‌یاب OBD یک استاندارد آمریکایی است. معادل این استاندارد در اروپا EOBD³ است. این استاندارد در سال 2006 توسط اتحادیه اروپا معرفی شد که همان OBDII بود با این تفاوت که پارامترهای بیشتری را تحت کنترل قرار می‌داد و برای خودروهایی با حداکثر 8 سرنشین یا خودرو با وزن کمتر از 2500 کیلوگرم اعمال می‌شود. ابزارهای مدرن عیب‌یابی خودرو EOBD نه تنها DTCها را نمایش می‌دهند، بلکه کدها را تفسیر و ترجمه می‌کنند، داده‌های سنسور را به صورت برخط نشان می‌دهند (پارامترها) و توصیه‌های تعمیر را به تعمیرکار ارائه می‌دهند.

نکته اینکه EOBDII⁴ نسل بعدی یا دوم EOBD نبوده و به ویژگی‌های خاص سازنده اشاره دارد که خودروسازان به ابزارهای OBDII و EOBD اضافه می‌کنند. EOBDII دسترسی به اطلاعات ها و پارامترهای اضافی را ممکن می‌سازد که در غیر این صورت با دستگاه‌های دیاگ و عیب‌یاب OBD2 یا EOBD به آنها دسترسی نداشتید. از آنجایی که دیاگ‌های EOBDII مخصوص سازنده خودرو هستند، فقط روی برندهای خودرویی که برای آنها طراحی شده‌اند کار می‌کنند.

استاندارد مورد استفاده در خودروهای ساخته شده توسط ژاپن به نام J-OBD شناخته می‌شود. این استاندارد به سیستم‌های OBDII گفته می‌شود که در ژاپن ساخته شده اما در ایالات متحده فروخته می‌شود. خودروسازهای ژاپنی تویوتا، هوندا، مزدا، میتسوبیشی، نیسان، سوزوکی و سوپارو از J-OBD بهره می‌برند.

³ European On-Board Diagnostics

⁴ 2nd generation European On-Board Diagnostics

استاندارد استرالیایی OBDII با نام ADR 79 شناخته شده و برای خودروهای سنگین و کامیون‌ها HD-OBD نامیده می‌شود.

در آینده‌ای نزدیک باید منتظر استاندارد OBDIII باشیم که انتظار می‌رود داده به‌صورت تلماتیک به مرکز فرستاده شده و در نتیجه خودرو به‌صورت آنلاین چک می‌شود!

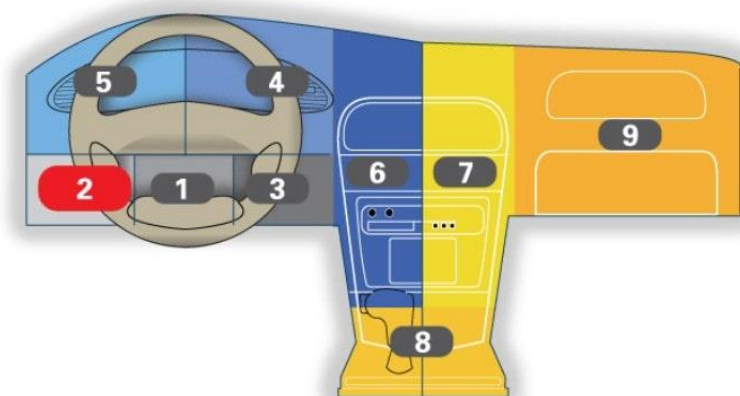
1-1- کاربرد پایه‌های OBDII

جدول 1- توضیح پایه های OBDII.

پین	توصیف	پین	توصیف
1	معمولاً نشان‌دهنده ACC	9	
2	J1850 BUS+	10	J1850 BUS-
3		11	BSI
4	زمین شاسی	12	ایر بگ و abs
5	زمین سیگنال	13	
6	CAN High	14	CAN Low
7	KLine	15	LLine
8		16	ولتاژ باتری

1-2- محل نصب پورت OBDII

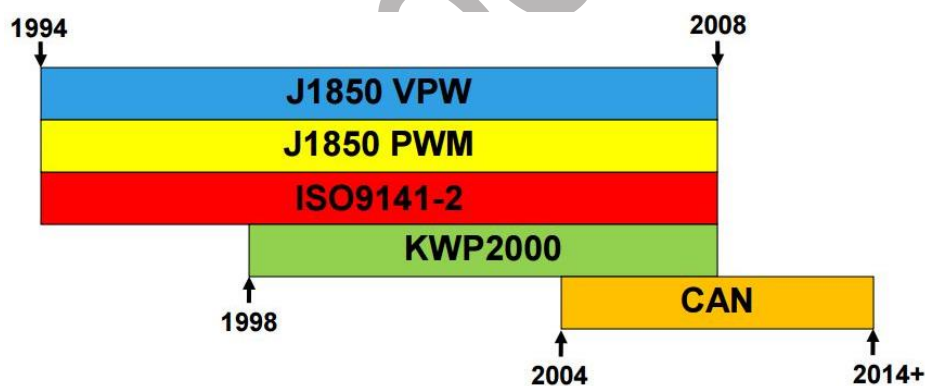
محل نصب پورت OBDII در خودرو در شکل 6 نشان داده شده است. همانطور که مشاهده می‌شود محل نصب کانکتور می‌تواند بسیار متفاوت باشد. البته عموماً در خودروها گزینه 1 یا 2 بسیار متداول هستند.



شکل 6 - محل نصب پورت OBDII در خودرو.

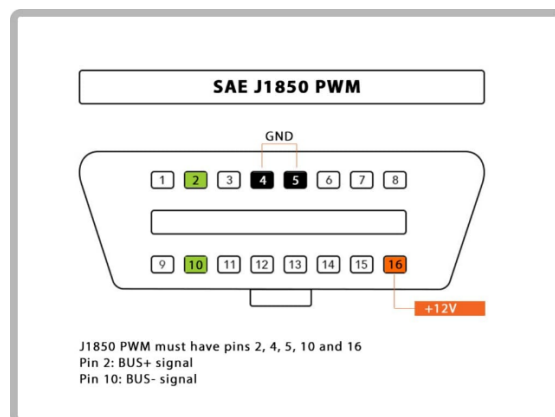
1-3- پروتکل‌های متداول در OBDII

به‌طور کلی 5 نوع پروتکل برای OBDII مورد استفاده قرار می‌گیرد. 1- SAE J1850 PWM 2- SAE J1850 3- VPW 4- ISO9141-2 5- ISO14230-4 (KWP2000) و 5- ISO 15765 CAN. تاریخچه استفاده از این پروتکل‌ها در شکل 7 ترسیم شده است.

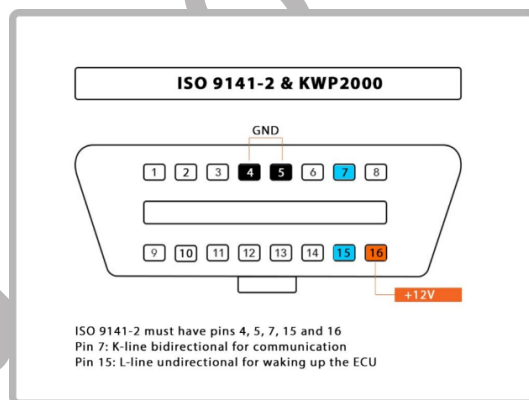
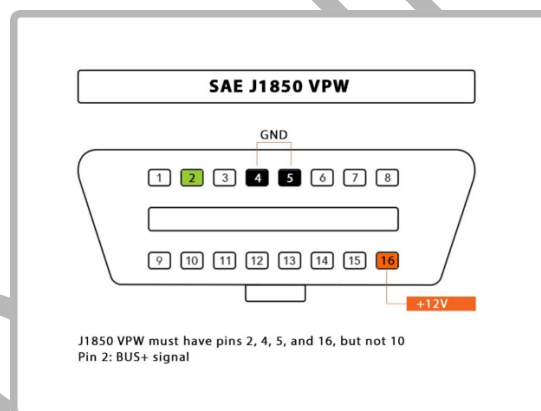


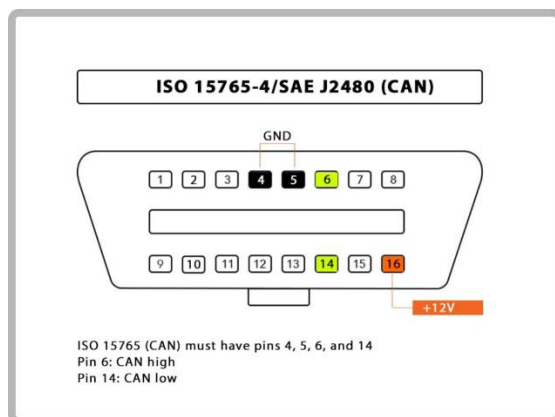
شکل 7- تاریخچه استفاده از پروتکل‌ها در OBDII.

کلاس SAE	B
سرعت انتقال داده‌ها	40.6kb/s
شرکت استفاده‌کننده	Ford
زمان استفاده	تا 2007
نوع شبکه	زوج سیم



کلاس SAE	B
سرعت انتقال داده‌ها	10.4kb/s
شرکت استفاده‌کننده	GM
زمان استفاده	تا 2007
نوع شبکه	تک سیم





1-4 کاربردهای OBD

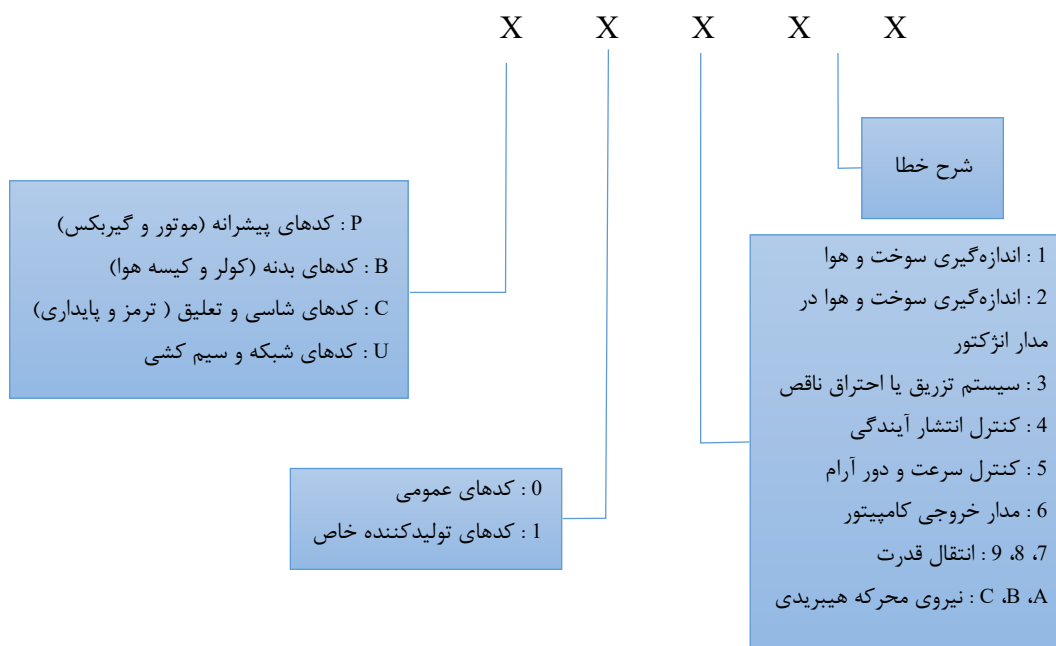
همان‌گونه که از نام OBD مشخص است، کاربرد اولیه این درگاه، تشخیص عیب می‌باشد. زمانی که حسگرهای خودرو نشان دهد که مشکلی رخ داده است، سنسورها پیغامی تحت عنوان “کد خطا” ایجاد می‌کنند. این خطا ممکن است به صورت “چراغ چک” یا برخی دیگر اخطارها روی داشبورد نمایان گردد. اسکنرهای OBD می‌توانند این عیب‌ها را بررسی و دقیقاً تعیین کنند چه مشکلی رخ داده است. این موارد به محض برطرف شدن از حافظه کامپیوتر خودرو حذف خواهد شد. به طور مثال، بروز ایراد “بسته نبودن درب باک بنزین” به صورت کد خطا نمایش داده می‌شود. Diagnostic Trouble Codes (DTS) یا همان کدهای تشخیص خطا، در سیستم کامپیوتری ذخیره می‌شوند. این کدها می‌توانند در بین تولیدکنندگان خودرو متفاوت باشند؛ بنابراین هرکسی می‌تواند با یک ابزار اسکن OBD-II به پورت متصل شود و کدهای عیب را از کامپیوتر بخواند.

1-5 معرفی کدهای خطا

هر واحد کنترل الکترونیکی مانند ECU دارای اجزای مختلفی مانند حسگرها (از جمله دمای مایع خنک‌کننده، دور موتور، اکسیژن، سرعت و...)، عملگرها (از جمله انژکتور، استپر موتور، رله پمپ بنزین و...) و... است. واحد کنترل الکترونیکی به طور پیوسته و مداوم برای اطمینان از عملکرد صحیح اجزای مختلف، آنها را بازرسی و چک می‌کند (بسته به میزان فناوری واحد الکترونیکی تعداد بازرسی‌ها به چند صدم ثانیه می‌رسد). اگر در هنگام بازرسی مشکلی مشاهده شود کد اشکال در عیب‌یابی آن DTC⁵ یا به اختصار کد خطای مربوطه را حافظه ذخیره می‌کند. ممکن است آن کد خطا چراغ هشدار (چراغ چک) را روشن نماید. با اتصال دستگاه عیب‌یاب به OBDII و در

⁵ Diagenetic Trouble Code

خواست کدهای خطا، کدهای خطای ذخیره شده در واحد الکترونیکی به دستگاه عیب‌یاب منتقل شده و تعمیرکار آنها را مشاهده می‌کند. کدهای خطای استاندارد معمولاً شامل ۵ کاراکتر ASCII است که به صورت نشان داده شده در شکل 8 تعبیر می‌شوند. کدهای خطا، راهنمای برای شناسایی مشکلی است که در خودرو رخ داده است. کدها باید به همراه مستندات راهنمای خودرو استفاده شوند چراکه ممکن است کد خطا در هر خودرو معنایی متفاوت داشته باشد. هر کد خطا در خیلی از موارد می‌تواند به دلایل مختلف ایجاد شود.



شکل 8 - شرح کد خطا.

1-5-1-1- دسته‌بندی کد خطا

1-5-1-1-1- کد خطای موقت (معلق/در انتظار)

کدهای معلق یا موقت یا در انتظار یک گزارش لحظه‌ای در موتور خودرو و یا هر یونیت دیگر خودرو است. به بیان ساده‌تر، یک نقص تصادفی می‌تواند در طول چرخه فعالیت فعلی خودرو ایجاد شود؛ اما فقط برای یک ثانیه این نقص اتفاق می‌افتد. این باعث می‌شود که خطای «موقت» ایجاد شود. معمولاً کدهای معلق خیلی شدید نیستند، اما نباید آنها را نادیده بگیرید. تعمیر و بررسی کدهای موقت باید پیگیری شوند. به عنوان مثال، ممکن است در جاده رانندگی کنید و به یک دست‌انداز برخورد کنید که باعث حرکت شمع یا وایرها شود. وایرها ممکن است به اشتباه تکان بخورد و باعث شود که شمع جرقه خوبی نداشته باشد.

نتیجه نهایی این است که ماشین یک لحظه موج زده و با لرزش کار کند و برای یک یا دو ثانیه احتراق به درستی انجام نمی‌شود و باعث ایجاد یک کد موقتی می‌شود. معمولاً در این صورت متوجه روشن شدن نشانگر هشدار یا

چراغ چک نخواهید شد. این به این دلیل است که از آنجایی که فقط یک بار و به طور خلاصه خطا اتفاق افتاده است، رایانه خودرو هیچ مشکلی را تشخیص نمی‌دهد و بنابراین هیچ کدی دائمی را نمایش نمی‌دهد.

کدهای خطای موقت اساساً یک نوع کد “تعمیرات پیشگیرانه” هستند. در واقع نشان‌دهنده مشکلی در خودرو نیستند تا زمانی که به طور متوالی اتفاق بیفتند.

2-1-5-1-2 کد خطای ذخیره‌شده

هنگامی که خطایی رخ می‌دهد اگر این خطا با استدلال نقص فنی تشخیص داده شود، کد خطای متناظر با آن که نشان‌دهنده اتفاق رخ داده هست را در رایانه ذخیره می‌شود و در نهایت چراغ هشدار روشن می‌شود. این کدها ممکن است در همه پلتفرم‌های خودرو یکسان باشند. این بدان معنی است که همه تولیدکنندگان خودرو تقریباً از همان “کد خطا” برای بسیاری از علائم مشابه استفاده می‌کنند. به عنوان مثال، اصطلاح P0420 معمولاً به معنای مشکل مبدل کاتالیزوری است و در بسیاری از خودروها به معنی اشکال در سیستم کاتالیزور گزارش می‌شوند.

به طور خلاصه، تفاوت بین کدهای دائم و موقت ساده است. یکی به باعث روشن شدن چراغ هشدار می‌شود؛ زیرا یک نگرانی مطمئن را تشخیص داده است. دیگری فقط به عنوان یک اقدام پیشگیرانه وجود خواهد داشت. با استفاده از آسان دیاگ شما می‌توانید این نوع کد را از هم تفکیک کنید

3-1-5-1-3 کد خطای جاری

خطای جاری خطایی است که در حال حاضر این خطا در خودرو وجود دارد و یونیت کنترل‌کننده و یا حس‌گر پیوسته گزارش خطا را ارسال می‌کند، به عبارت دیگر پس از تعداد معینی از سفر خطا دوباره رخ می‌دهد. در این صورت خطا به عنوان خطای جاری گزارش می‌شود. در صورت پاک‌سازی خطای جاری بدون برطرف کردن نقص، همچنان خطا مشاهده خواهد شد.

4-1-5-1-4 کد خطای غیرجاری

در صورتی که خطا توسط حس‌گرهای کنترل‌کننده خودرو گزارش نشود کامپیوتر خودرو خطا را به وضعیت غیرجاری تغییر می‌دهد. پاک‌سازی خطا معمولاً خطای غیرجاری را از بین می‌برد. کد خطای غیرجاری خیلی خطرناک نبوده اما نیاز به بررسی جهت پیشگیری و جلوگیری از رخ دادن مجدد دارد.

2- معرفی پروتکل KLine

پروتکل K-Line یک روش استاندارد ارتباطی OBDII هست که قبل از آنکه پروتکل CAN فراگیر شود در خودروها استفاده می‌شده است. K-Line فقط یک اسم فانتزی برای UART تک سیم با ارتباط نیمه دوطرفه⁶ که

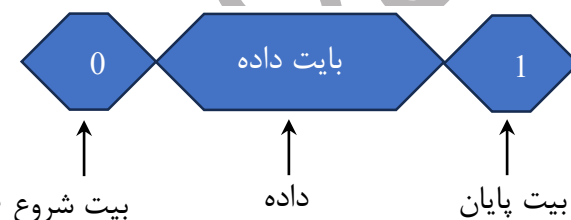
⁶ half-duplex

دارای سرعت 10.4 kbps بوده و از سطوح ولتاژ 0 و 12 ولت استفاده می کند، است. سطح ولتاژ بالای آن در حقیقت همان ولتاژ باتری است که بین 12 تا 14/4 (زمان روشن بودن موتور) تغییر می کند.

2-1- بسته استاندارد داده در Kline

پروتکل KLine برای تبادل داده از استانداردهای UART پیروی می کند. به این صورت که در حالتی که هیچ تبادل داده‌ای در حال انجام نباشد خط انتقال در حالت بیکاری⁷ قرار داشته و خط ارسال در سطح ولتاژ یک منطقی است. برای شروع ارسال داده خط از مقدار صفر منطقی تغییر داده می شود این صفر شدن به مدت یک بیت باید طول بکشد و به آن بیت شروع⁸ گفته می شود. در ادامه یک بایت داده (معادل هشت بیت) از سمت کم ارزش ترین بیت ارسال می شود. در نهایت یک بیت پایان⁹ که معادل یک منطقی است ارسال می شود. طبق آنچه گرفته شد برای تبادل داده با KLine باید از استاندارد UART با 8 بیت داده، بدون بیت توازن و یک یا دو بیت پایان (اختیاری!) استفاده کرد (

شکل 9).



شکل 9- بسته استاندارد داده در KLine.

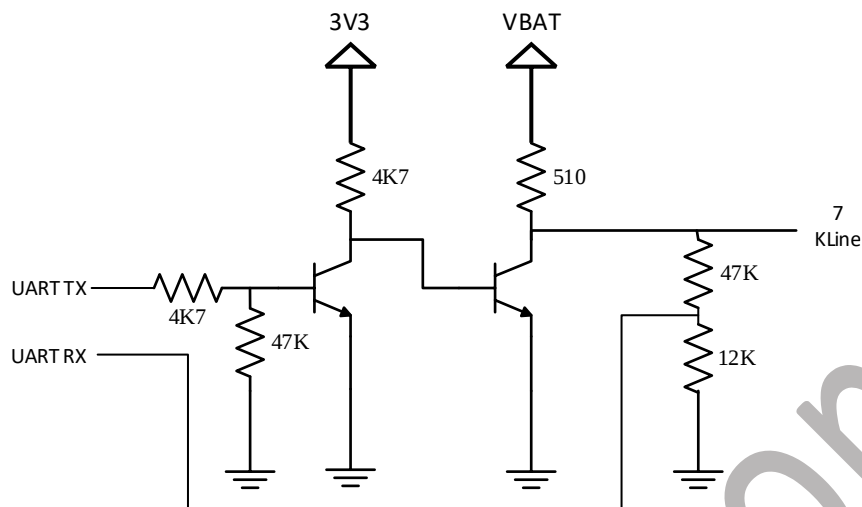
2-2- سخت افزار لازم برای اتصال به میکروکنترلر

لایه سطح فیزیکی KLine بسیار ساده بوده و برای ارتباط با این روش به یک میکروکنترلر و چند ترانزیستور نیاز هست (شکل 10).

⁷ Idle

⁸ Start bit

⁹ Stop bit



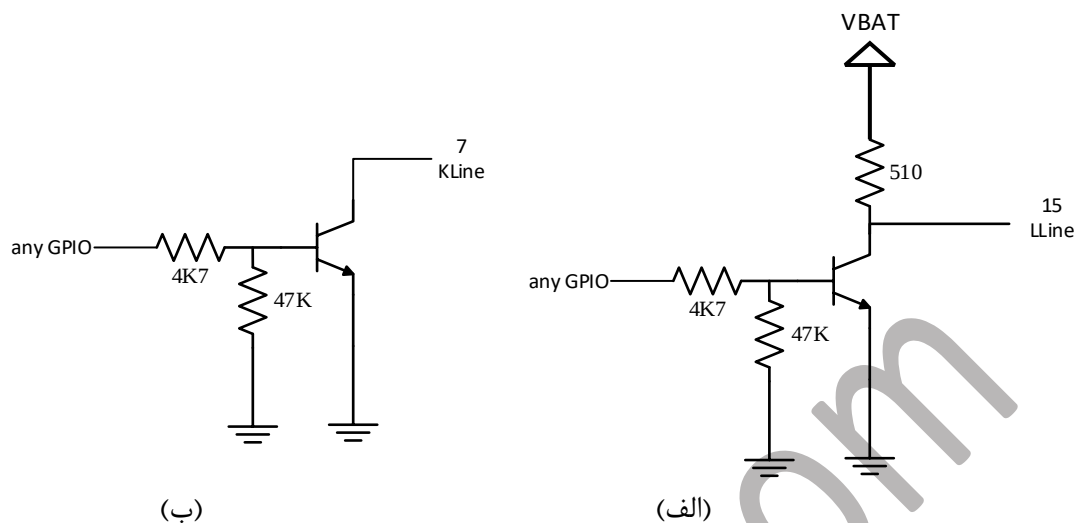
شکل 10- مدار مبدل KLine به UART.

دریافت داده از K-Line با استفاده از یک مقسم ولتاژ ساده انجام می‌شود که باتوجه به ولتاژ کاری میکروکنترلر طراحی می‌گردد. باتوجه به اینکه حداکثر ولتاژ K-Line بین 12 تا 14/4 تغییر می‌کند، خروجی مقسم نباید از ولتاژ کاری میکروکنترلر بیشتر شود.

ارسال داده نیز بسیار ساده هست. تنها نکته این است که سطح حالت بیکاری ولتاژ UART باید بالا باشد؛ بنابراین در اینجا باید از دو عدد ترانزیستور NPN سری با آرایش کلکتور باز و مقاومت بالا کشنده استفاده کرد. در صورتی که میکروکنترلر (مانند AVR XMEGA) توانایی معکوس کردن پایه TX را داشته باشد می‌توان از یک ترانزیستور هم استفاده کرد.

خط LLine فقط از میکروکنترلر ارسال شده و طریقه اتصال این پایه به میکروکنترلر در شکل 11 (الف) ترسیم شده است.

باتوجه به اینکه الگوی شروع کند باید با نرخ ارسال 5 bps ارسال شود و معمولاً این نرخ داده به سختی با UART در میکروکنترلرها حاصل می‌شود بهتر است از یک مدار کنترلی دیگر شکل 11 (ب) برای ارسال الگو استفاده شود.



شکل 11- مدار اتصال پین LLine و پین کنترل کننده KLine.

3- قالب بندی داده ها در K-Line

3-1 قالب بندی پیام درخواست

همان طور که در شکل 12 نشان داده شده، ساختار پیام شامل سه قسمت است: هدر، داده، چک سام. به عنوان مثال برای خوانند مقدار دمای آب خنک کننده کد {0xC2,0x33,0xF1,0x01,0x05,0xEC} ارسال می شود.

چک سام	داده	شناسه سرویس	طول	آدرس منبع	آدرس هدف	فرمت
چک سام (یک بایت)	بخش داده (حداکثر 255 بایت)	بخش هدر (حداکثر 4 بایت)				

(1) بایت های اختیاری بسته به بایت فرمت

شکل 12 - ساختار پیام در KLine

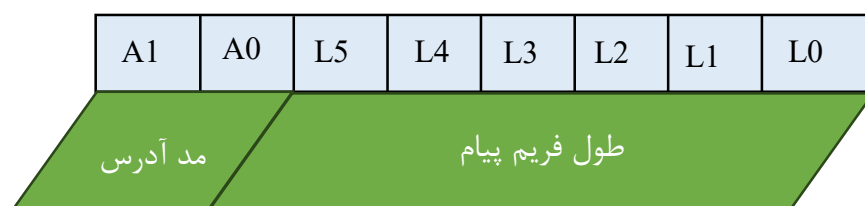
3-1-1- بخش هدر

این بخش می‌تواند حداکثر شامل 4 بایت باشد. اولین بایت، بایت فرمت بوده که شامل اطلاعاتی در مورد فرمت پیام است. بایت‌های دوم و سوم که نشان دهنده‌ی آدرس منبع و هدف هستند می‌توانند وجود داشته باشند یا نباشند. در نهایت بسته به طول داده، بایت چهارم می‌تواند در هدر وجود داشته باشد یا نباشد. مثلاً در برای درخواست دمای آب خنک کننده بخش هد شامل {0xC2, 0x33, 0xF1} است که 0xC2 بایت فرمت، 0x33 بایت آدرس هدف و 0xF1 بایت آدرس منبع بوده و بایت طول در آن وجود ندارد.

3-1-1-1- بایت فرمت

دو بیت با ارزش بایت فرمت، مد آدرس دهی را مشخص می‌کند و 6 بیت دیگر طول فریم پیام را مشخص می‌کند (شکل 13). حالت‌های مختلف آدرس دهی در جدول 2 آورده شده است. اگر دو بیت مربوطه "00" باشند هیچ اطلاعات آدرسی ارسال نمی‌شود. اگر دو بیت مربوطه "01" باشند آنگاه در مد CRAB است. در صورتی که دو بیت مربوطه "10" باشند آنگاه آدرس دهی به صورت فیزیکی است. یعنی فقط به یک یونیت خاص درخواست داده می‌شود. و در نهایت اگر "11" باشد، درخواست به گروهی از یونیت‌ها داده می‌شود.

بیت‌های L5-L0 طول داده موجود در پیام را مشخص می‌کنند (بایت چک سام شامل نمی‌شود)؛ بنابراین طول داده می‌تواند بین 0 تا 63 بایت باشد. اگر این بیت‌ها صفر باشند از بایت چهارم در هدر برای نشان دادن طول داده با حداکثر 255 استفاده می‌شود. در مثال خواندن دمای آب خنک کننده، بایت فرمت 0xC2 بود. یعنی مد آدرس دهی برابر با آدرس دهی فانکشنال و تعداد بایت داده ارسالی برابر با 2 است.



شکل 13- ساختار بایت قالب بندی در Kline

جدول 2- انواع مد آدرس‌دهی در KLine

A1	A0	مد
0	0	بدون اطلاعات آدرس

0	1	مد CRAB، در این مد از بایت فرمت 68 و 48 استفاده می‌شود.
1	0	با آدرس‌دهی فیزیکی
1	1	با آدرس‌دهی فانکشنال

2-1-3- بایت آدرس هدف

این بایت آدرس هدف پیام را مشخص می‌کند و همیشه همراه با آدرس منبع استفاده می‌شود. هم به صورت فیزیکی و هم فانکشنال می‌تواند باشد. آدرس فیزیکی می‌تواند بایت آدرس در شروع کند باشد یا طبق استاندارد بین‌المللی (شکل 14). این بایت اختیاری بوده و فقط در باس‌هایی با توپولوژی چند نود ضروری است. برای ارتباطات نود به نود (تک نود) می‌توان از این بایت صرف نظر کرد. در مثال خواندن دمای آب خنک کننده، مقدار بایت آدرس هدف، 0x33 است.

3-1-3- بایت آدرس منبع

این بایت آدرس دستگاه فرستنده را مشخص می‌کند. این آدرس باید یک آدرس فیزیکی باشد و مقدار آن مشابه مقدار بایت آدرس هدف تعیین می‌شود. این بایت اختیاری بوده (همیشه با آدرس هدف همراه است) و فقط در شبکه‌های چند نودی ضروری است. در شبکه‌های تک نود می‌توان از آن صرف نظر کرد. در مثال خواندن دمای آب خنک کننده، مقدار بایت آدرس هدف، 0xF1 است. محدود تعریف شده این آدرس در شکل 14 آورده شده است.

شکل 14- آدرس ECU و درخواست کننده برای KWP

مقدار hex	آدرس پیشرانه (قوای محرکه)
00 – 0F	شرکت سازنده
10 – 17	موتور
18 – 1F	انتقال قدرت (گیربکس، دیفرانسیل)
	شاسی
20 – 27	شرکت سازنده
28 -2F	ترمز
30 – 37	فرمان

38 – 3F	تعلیق
	بدنه
40 – 57	شرکت سازنده
58 – 5F	ایمنی
60 – 6F	نمایشگر
70 – 7F	روشنایی
80 – 8F	صوت و سرگرمی
90 – 97	ارتباط شخصی
98 – 9F	سرمایش و گرمایش
A0 – BF	کنترل درب، صندلی، پنجره و...
C0 – C7	امنیت
C8 – CF	توسعه در آینده
D0 – EF	سازند
F0 – FD	دستگاه عیب یاب
FE	همه نودها
FF	هیچ نودی

4-1-3- بایت طول پیام

اگر طول پیام در بایت فرمت داده صفر باشد، از این بایت برای مشخص کردن طول پیام استفاده می‌شود. این بایت امکان ارسال داده‌هایی با طول بیش‌تر از 63 بایت را فراهم می‌کند. برای داده‌هایی با طول‌های کمتر می‌توان از آن صرف نظر کرد. این بایت مقدار طول پیام از ابتدای بخش داده تا چک سام (شامل چک سام نیست) را مشخص می‌کند. بنابراین طول داده می‌تواند بین 1 تا 255 باشد. برای طول داده کمتر از 64 بایت دو امکان وجود دارد. یک استفاده از بایت فرمت یا استفاده از بایت طول پیام. البته لزومی وجود ندارد که یک واحد از این هر دو روش به صورت هم‌زمان پشتیبانی کند. در جدول 3- چگونگی تعیین طول پیام به طور خلاصه آورده شده است. در مثال خواندن دمای آب خنک کننده، بایت طول پیام وجود ندارد.

جدول 3- چگونگی تعیین طول پیام در KLine

بایت طول داده	بایت فرمت	طول داده
---------------	-----------	----------

وجود دارد	XX00 0000	کمتر از 64
وجود ندارد	XXLL LLLL	کمتر از 64
وجود دارد	XX00 0000	بیشتر از 64

با توجه با آنچه در بالا بیان شد، چهار نوع متفاوت از پیام وجود دارد

1- هدر بدون بایت‌های آدرس و طول داده

چک سام	داده	شناسه سرویس	فرمت
--------	------	-------------	------

شکل 15 - ساختار پیام در KLine بدون بایت های آدرس و طول.

2- هدر با بایت های آدرس و بدون بایت طول داد

چک سام	داده	شناسه سرویس	آدرس منبع	آدرس هدف	فرمت
--------	------	-------------	-----------	----------	------

شکل 16 - ساختار پیام در KLine با بایت های آدرس ولی بدون بایت طول.

3 - هدر بدون بایت های آدرس و با بایت طول داده

چک سام	داده	شناسه سرویس	طول	فرمت
--------	------	-------------	-----	------

شکل 17 - ساختار پیام در KLine بدون بایت های آدرس و با بایت طول.

- هدر با بایت‌های آدرس و طول داده

چک سام	داده	شناسه سرویس	طول	آدرس منبع	آدرس هدف	فرمت
--------	------	-------------	-----	-----------	----------	------

شکل 18 - ساختار پیام در KLine با بایت های آدرس و بایت طول.

5-1-3- بایت‌های داده

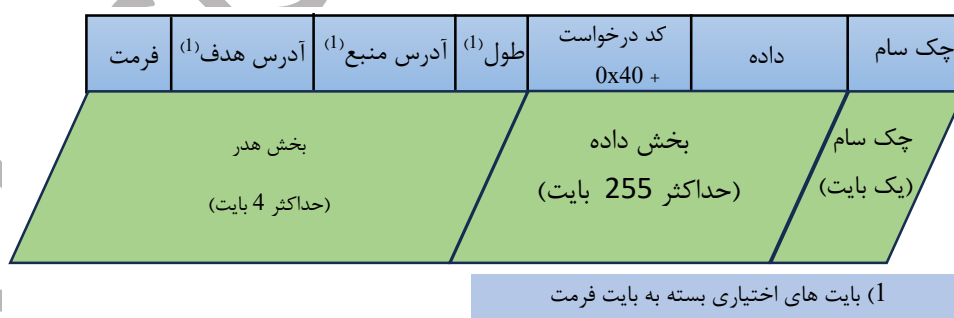
بخش داده می‌تواند بسته به استفاده از بایت طول داده حداکثر شامل 64 یا 255 بایت باشد. اولین بایت شناسه سرویس است. با توجه به سرویس انتخاب شده، بایت های بعدی می‌تواند پارامتر یا داده باشند. در مثال خواندن دمای آب خنک کننده، بایت‌های داده برابر با {0x01,0x05} هستند.

3-1-6- بایت چک سام

4- این بایت به انتهای پیام اضافه می‌شود. برای محاسبه‌ی آن کافی است که همه بایت‌های موجود در پیام را با هم جمع کرد. در مثال خواندن دمای آب خنک کننده، مقدار بایت چک سام برابر با $0xC0 = 0x05 + 0x01 + 0xF1 + 0x33 + 0xC2$ است.

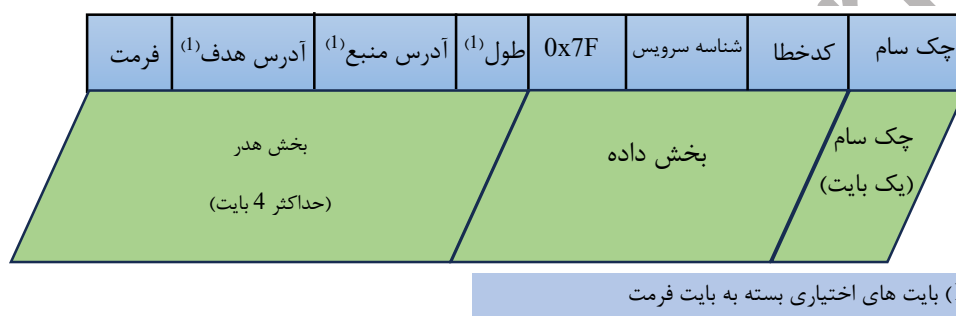
5- قالب بندی پیام درخواست

پاسخ ECU به درخواست ارسال شده به آن می‌تواند مثبت یا منفی باشد. ساختار پیام پاسخ مثبت ECU به درخواست در شکل 19 نشان داده شده است. در پیام پاسخ ECU، بیت 6ام از کد درخواست ارسال شده را یک می‌کند به عبارت دیگر کد درخواست ارسال شده را با $0x40$ جمع می‌کند و به عنوان اولین داده در پیام قرار می‌دهد. به عنوان مثال ECU پیام پاسخ $\{0x83, 0xF1, 0x11, 0x41, 0x05, 0x76, 0x41\}$ را در جواب درخواست $\{0xC2, 0x33, 0xF1, 0x01, 0x05, 0xEC\}$ ، کد درخواست دمای مایع خنک کننده، ارسال می‌کند. در ادامه بخش داده، ادامه داده‌های ارسالی که در اینجا $0x05$ است ارسال می‌شود و در نهایت جواب که در اینجا بایت $0x76$ است ارسال می‌شود که داده خام است و برای تعیین دمای آب باید آنرا در فرمول مربوط به آن قرار دهیم. مثلاً در اینجا باید از فرمول A-40 استفاده کرد که همان داده خام است. در نتیجه دما آب برابر با 78 درجه سانتی‌گراد محاسبه می‌شود.



شکل 19- ساختار پیام پاسخ مثبت ECU به یک درخواست در ارتباط KLine.

اگر ECU به هر دلیلی نتواند پاسخ مثبت به درخواست بدهد پاسخ ارسالی به صورت شکل 20 خواهد بود. که در آن اولین بایت از قسمت داده برابر با 0x7F است که نشان دهنده پاسخ منفی به درخواست است. سپس شناسه سرویس درخواست ارسالی مثلاً در اینجا 0x01 ارسال می‌شود. در نهایت کد خطا که نشان دهنده علت پاسخ منفی به درخواست است را ارسال می‌کند. به عنوان مثال کد خطا 0x78 به معنای این است که ECU درخواست را به دستی دریافت کرده است اما برای پاسخ به این درخواست به زمان بیشتر از زمان قراردادی نیاز دارد. در جدول 4 کد خطاهای ارسالی لیست شده‌اند.



شکل 20- ساختار پیام پاسخ منفی ECU به یک درخواست در ارتباط KLine.

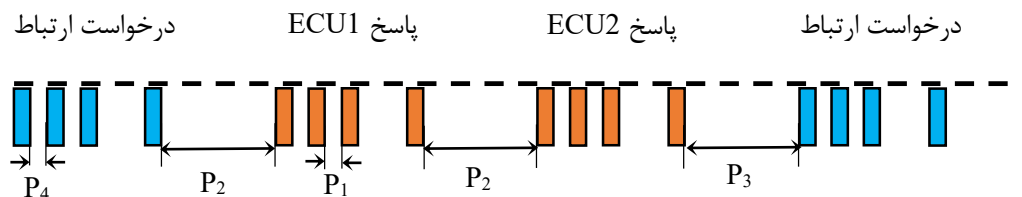
جدول 4- انواع کدهای خطا در ارتباط KLine.

کد خطا hex	توصیف
10	رد کردن کلی: این کد خطا زمانی ارسال می‌شود که خطای رخ داده در لیست خطاهای ممکن نباشد.
11	درخواست پشتیبانی نمی‌شود: این خطا زمانی رخ می‌دهد که سرویس درخواستی توسط واحد الکترونیکی پشتیبانی نشود.
12	زیرسرویس پشتیبانی نمی‌شود - قالب نامعتبر: این خطا زمانی رخ می‌دهد که زیر سرویس درخواستی توسط واحد الکترونیکی پشتیبانی نشود یا فرمت آن درست نباشد.
21	سرور مشغول است الان نمی‌تواند جواب دهد مجدد بعداً تلاش کنید: این خطا بیان می‌کند که واحد الکترونیکی در حال حاضر مشغول به انجام کار دیگری است و نمی‌تواند به سرویس درخواستی جواب دهد. این بدین معنا است که در زمان دیگری واحد الکترونیکی می‌تواند پاسخ دهد.
22	شرایط درست نیست یا ترتیب در درخواست رعایت نشده: این خطا بیان می‌کند که ترتیب سرویس درخواستی درست نیست. این خطا برای درخواست‌هایی رخ می‌دهد که چند پیام باید به ترتیب خاصی ارسال شوند و در صورتی که ترتیب رعایت نشود این خطا رخ می‌دهد.
23	روتین کامل نشده است.

31	درخواست خارج از محدوده است: این خطا بیان می‌کند که مقدار ارسال شده از مقدار حداکثر مجاز بیشتر است مثلاً 111 فرستاده شده در صورتی که حداکثر مقدار ممکن 100 است.
33	دسترسی امکان‌پذیر نیست: این خطا بیان می‌کند که جنبه‌های امنیتی برای سرویس درخواستی رعایت نشده است.
35	کلید نامعتبر: این خطا زمانی رخ می‌دهد که کلید ارسال از دستگاه دیاگ با کلید ذخیره شده در واحد الکترونیکی یکسان نباشد برای سرویس 27
36	تعداد تلاش‌ها از مقدار مجاز بیشتر شده: این خطا زمانی رخ می‌دهد که تعداد دفعات ارسال کلید از تعداد مجاز بیشتر شده باشد سرویس 27
37	تأخیر موردنیاز به اتمام نرسیده
40	دانلود مورد پذیرش قرار نگرفته
41	نوع دانلود متناسب نیست
42	امکان دانلود از آدرس مشخص شده نیست
43	تعداد بایت درخواست شده امکان دانلود را ندارند
50	درخواست آپلود مورد قبول واقع نشد
51	نوع آپلود متناسب نیست
52	در آدرس مشخص شده امکان آپلود نیست
53	تعداد بایت درخواست شده امکان آپلود را ندارند.
71	انتقال در حالت معلق است
72	انتقال لغو شد
74	آدرس بلاک برای انتقال نامعتبر است
75	تعداد بلاک برای انتقال نامعتبر است
76	نوع انتقال نامعتبر است
77	چک سام داده‌های منتقل شده نامعتبر است
78	درخواست به‌درستی دریافت شده اما برای پاسخ نیاز به زمان بیشتری است.
79	تعداد بایت‌ها در عملیات انتقال درست نیست
80-FF	

6- زمان‌بندی در ارتباط KLine

زمان‌بندی ارتباط KLine در حالت معمول به صورت شکل 21 است. توصیف زمان‌های P₁-P₄ در جدول 5 آورده شده است.



شکل 21 - زمان بندی و جریان داده در پروتکل KLine.

جدول 6 - توصیف پارامترهای زمان بندی در KLine.

مقدار	توصیف
P ₁	زمان بین بایت‌ها در پیام پاسخ ECU
P ₂	مدت زمانی که طول می‌کشد تا ECU به یک درخواست جواب دهد.
P ₃	زمان بین پایان پاسخ ECU و شروع درخواست جدید
P ₄	زمان بین بایت‌ها در پیام درخواست

در حالت کلی دودسته زمان‌بندی قراردادی برای برقراری ارتباط تعریف شده است.

1- یک دسته برای ارتباط متعارف با آدرس‌دهی فیزیکی و فانکشنال.

2- دسته دیگر فقط برای ارتباط با آدرس‌دهی فیزیکی بوده که ارتباط سریع‌تر را فراهم می‌کند.

محدودیت‌های زمان‌بندی ارتباط توسط ECU با ارسال بایت KB1 در زمان شروع ارتباط اطلاع داده می‌شود (بخش...). البته ممکن است بتوان زمان‌بندی را با سرویس "دسترسی به پارامترهای زمان‌بندی" تغییر داد.

در جدول 7 و جدول 8 پارامترهای زمان‌بندی که به صورت قراردادی مورد استفاده قرار می‌گیرند آورده شده است.

جدول 7- مقادیر متعارف پارامترهای زمان‌بندی برای آدرس‌دهی فیزیکی و فانکشنال.

پارامتر زمان‌بندی	حداقل مقدار (ms)			حداکثر مقدار (ms)		
	حد پایین	قراردادی	گام	حد بالا	قراردادی	گام
P ₁	0	0	-	20	20	-
P ₂	0	25	0.5	(1)	50	(1)

P ₃	0	55	0.5	∞	5000	250
P ₄	0	5	0.5	20	20	-

جدول 8- مقادیر متعارف پارامترهای زمان بندی برای آدرس دهی فیزیکی.

پارامتر زمان بندی	حداقل مقدار (ms)			حداکثر مقدار (ms)		
	حد پایین	قراردادی	گام	حد بالا	قراردادی	گام
P ₁	0	0	-	20	20	-
P ₂	0	0	0.5	(1)	100	(1)
P ₃	0	0	0.5	∞	5000	250
P ₄	0	5	0.5	20	20	-

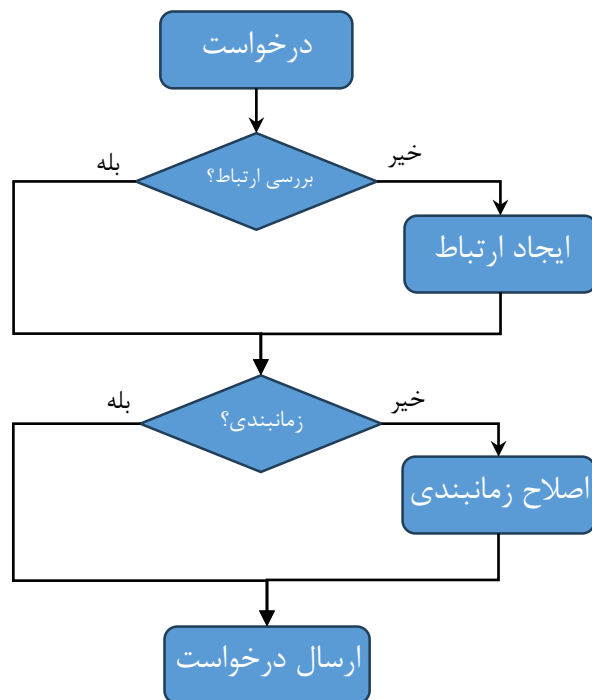
(1) اگر مقدار پارامتر زمان بندی متناسب با P_{2max} ارسال شده توسط ECU در جواب درخواست دسترسی به پارامترهای زمان بندی بین 0xF0 – 0x01 باشد گام آن 25 میلی ثانیه است مقدار P_{2max} آن برابر است حاصل ضرب مقدار در گام. اگر مقدار آن بین 0xF1-0xFE باشد مقدار P_{2max} برابر است با مقدار چهاربیت پایین ضربدر 256 ضربدر گام (P_{2max} = 0x0A * 256 * 25 = 6400 در صورتی که داشته باشم 0xFA).

6-1- مستثنات زمان بندی

در بعضی از مواقع در صورت لزوم ECU می تواند درخواست افزایش مقدار زمان P_{2max} برای ارسال پاسخ درخواست را ارسال کند. بدین منظور ECU یک یا چند پاسخ منفی با کد خطای 0x78 را ارسال می کند. این کد خطا فقط زمانی ارسال می شود که ECU نتواند در زمان مقرر پاسخ مثبت یا منفی به درخواست ارسال شده را بدهد. این درخواست افزایش زمان پاسخ در ECU و درخواست کننده اعمال می شود. در این زمان درخواست کننده همچنان منتظر دریافت پاسخ ECU می ماند و درخواست خود را مجدد ارسال نمی کند. در طرف دیگر ECU هم به محض اینکه درخواست ارسال شده را انجام داد، پاسخ مثبت یا منفی (با کد خطایی متفاوت از 0x78) خود را ارسال می کند. سپس هر دوی ECU و درخواست کنند مقدار زمان P_{2max} را به حالت پیش فرض برمی گردانند.

7- تبادل داده در KLine

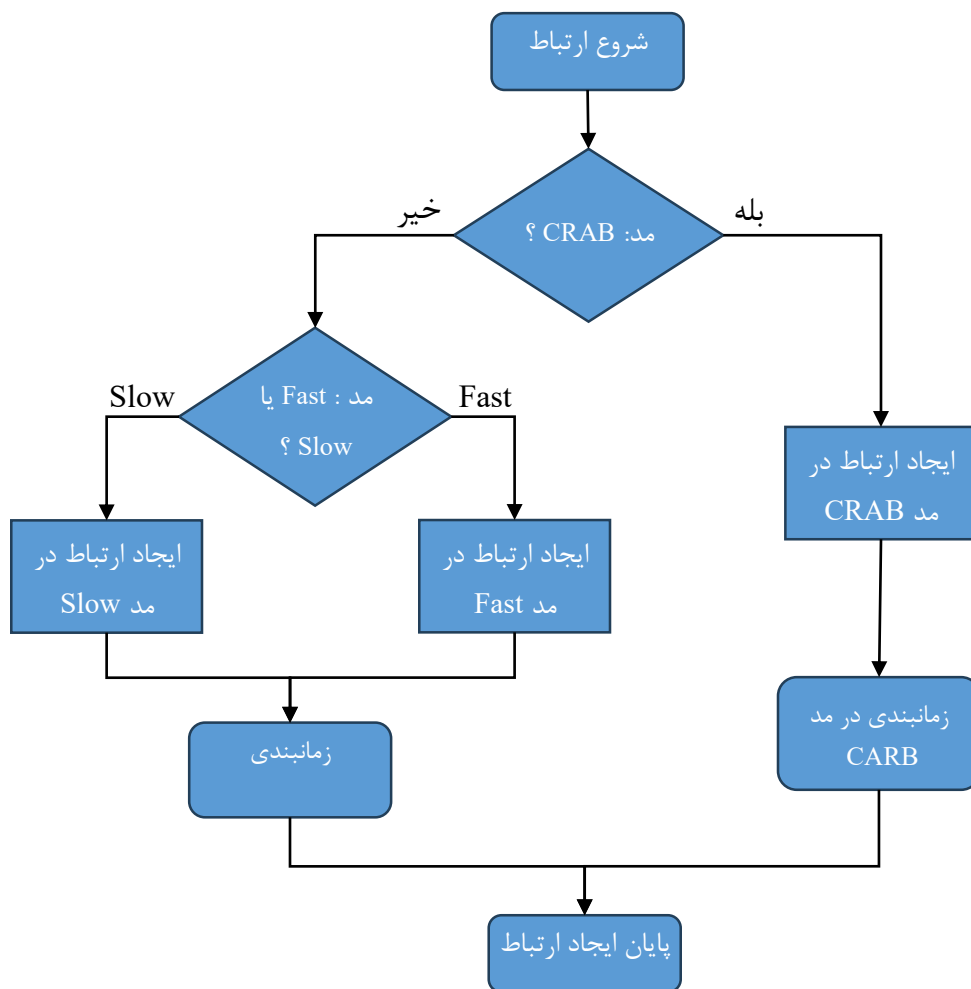
به منظور ارسال هر نوع درخواست، ابتدا باید ارتباط شکل بگیرد و زمان بندی به صورت صحیح پیاده سازی شده باشد. این موضوع در شکل 22 نشان داده شده است.



شکل 22- رویه شروع درخواست در KLine.

7-1- رویه شروع ارتباط

درخواست شروع ارتباط برای ایجاد ارتباط بر بستر Kline مورد استفاده قرار می گیرد. سه حالت ممکن برای ایجاد ارتباط وجود دارد. الگوریتم پیشنهادی برای ایجاد ارتباط در شکل 23 آورده شده است.

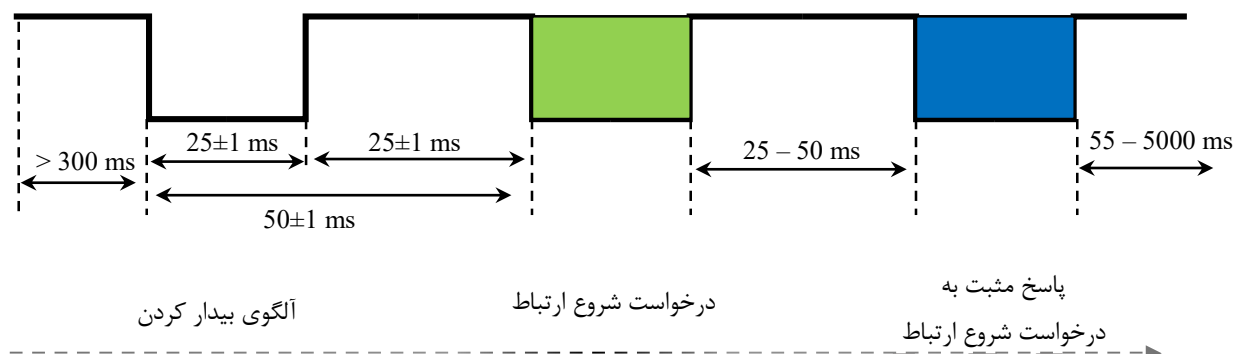


شکل 23- الگوریتم ایجاد ارتباط در KLine.

7-2- ایجاد ارتباط سریع

برای شروع سریع یا Fast init، یک الگوی Wake up به صورت نشان داده شده در شکل 24 ارسال می شود. دقت شود که زمان های 25 ms حداکثر ± 1 ms باید خطا داشته باشد و کل 50 ms باید ± 1 ms خطا داشته باشد. سپس بلافاصله یک فریم داده حاوی درخواست شروع ارتباط به واحد مربوطه فرستاده می شود. آن واحد باید یک پاسخ ارسال کند که در صورتی که پاسخ معتبر باشد به معنای آن است که شروع ارتباط موفقیت آمیز بوده است. در شکل 25 ساختار پیام درخواست برای شروع ارتباط نشان داده شده است. اگر آدرس دهی به صورت فیزیکی باشد بایت فرمت برابر با 0x81 خواهد بود (با توجه به اینکه طول داده ارسالی یک است $0x80 + 0x01$) در غیر این صورت 0xC1 خواهد بود. دو بایت بعدی آدرس هدف و منبع هستند و در نهایت بایت 0x81 که کد سرویس درخواست شروع ارتباط برای KLine است. بایت پایانی چک سام بایت های پیام است. به عنوان مثال، برای

درخواست شروع ارتباط با آدرس دهی فانکشنال ساختار پیام به صورت $\{0xC1, 0x33, 0xF1, 0x81, 0x66\}$ است. که در آن $0x33$ آدرس هدف فانکشنال، $0xF1$ آدرس منبع و $0x66 = 0xC1 + 0x33 + 0xF1 + 0x81$ است.



شکل 24- زمان بندی شروع سریع در KLine.

81 or C1	xx	xx	81	xx
بایت فرمت آدرس دهی فیزیکی: 81 آدرس دهی گروهی: C1	بایت آدرس هدف	بایت آدرس منبع	سرویس درخواست شروع ارتباط	چک سام

شکل 25- ساختار پیام درخواست شروع ارتباط در KLine.

ECU پس از دریافت درخواست شروع ارتباط در صورتی که شرایط درست باشد به صورت شکل 26 پاسخ معتبر را ارسال می کند. بایت طول پیام در اینجا ممکن است در متن پیام باشد یا نباشد. بایت $0xC1$ نشان دهنده پاسخ مثبت ECU به درخواست است. همانطور که قبلا بیان شد ECU با یک کردن بیت شماره 6ام کد درخواست که اینجا $0x81$ است این بایت را درست می کند که معادل این است که $0x81$ را با $0x40$ جمع کنیم.

xx	xx	xx	xx	C1	xx	xx	xx
بایت فرمت	بایت آدرس هدف	بایت آدرس منبع	بایت طول پیام	پاسخ مثبت به سرویس درخواست شروع ارتباط	KB1	KB2	چک سام

شکل 26- ساختار پیام جواب مثبت در KLine.

دو بایت بعدی بایت‌های کلیدی¹⁰ (KB1 و KB2) نامیده می‌شوند. این بایت‌های اطلاعاتی در مورد تعداد بایت‌های هدر، زمانبندی و طول پیام فراهم می‌کنند. در اینجا بایت KB2 معمولاً ثابت و برابر با 0x8F است. محتوای بایت KB1 در شکل 27 نمایش داده شده است. با ارزش‌ترین بیت، بیت توازن است. بیت بعدی همیشه یک است. دو بیت بعدی نوع زمانبندی پشتیبانی شده (متعارف یا سریع) را مشخص می‌کند. بیت‌های بعدی تعداد بایت‌های هدر را مشخص می‌کند و در نهایت دو بیت بعدی جایگاه طول داده در پیام را مشخص می‌کند (در بایت فرمت یا بایت طول یا هر دو). توضیحات کامل در جدول 9 و جدول 10 آورده شده است. به عنوان مثال ECU در پاسخ به درخواست بیان شده در بالا جواب {0x83,0xF1,0x11,0xC1,0xEF,0x8F,0xC4} را ارسال می‌کند. با توجه به متن پیام مشخص می‌شود که طول داده برابر با 3 بایت است (0x83)، جواب درخواست مثبت است (0xC1) و دو بایت کلیدی برابر با KB1 = 0xEF و KB2 = 0x8F هستند و در نهایت بایت چک سام برابر با 0xC4 هست. با توجه به مقدار بایت KB1 و توضیحات جدول 9 و جدول 10 می‌توان بیان کرد که این ECU از زمانبندی متعارف پشتیبانی کرده و از هدر یک بایتی (بدون بایت‌های آدرس) و هدر با بایت‌های آدرس پشتیبانی کرده و همچنین طول داده را هم می‌توان در بایت فرمت نوشت هم در بایت طول داده.

Parity	1	TP1	TP0	HB1	HB0	AL1	AL0
--------	---	-----	-----	-----	-----	-----	-----

شکل 27- ساختار بایت KB1 در KLine.

جدول 9- مقادیر ممکن برای بایت KB1 در KLine.

مقدار بیت	0	1
AL0	مقدار طول داده پیام در بایت فرمت پشتیبانی نمی‌شود.	مقدار طول داده پیام در بایت فرمت پشتیبانی می‌شود.
AL1	بایت طول پیام پشتیبانی نمی‌شود.	بایت طول پیام پشتیبانی می‌شود.
HB0	هدر یک بایتی پشتیبانی نمی‌شود.	هدر یک بایتی پشتیبانی می‌شود.
HB1	بایت‌های آدرس هدف و منبع پشتیبانی نمی‌شود.	بایت‌های آدرس هدف و منبع پشتیبانی می‌شود.
TP0 ^(*)	زمان‌بندی متعارف.	زمان‌بندی بسط داده شده.

¹⁰ KeyByte

زمان بندی متعارف.	زمان بندی بسط داده شده.	TP1(*)
-------------------	-------------------------	--------

(*) فقط حالت های $TP0, TP1 = 0, 1$ و $TP0, TP1 = 1, 0$ مجاز هستند.

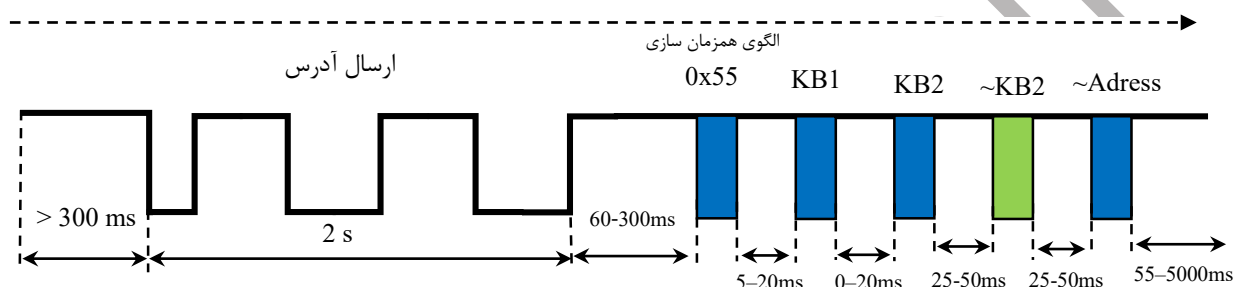
جدول 10-حالات ممکن برای بایت KB1.

زمان بندی	نوع هدر	بایت طول داده	KB1	
زمان بندی توسعه داده شده	هدر یک بایتی	بایت فرمت	0b1101 0101	0xD5
		بای طول داده	0b1101 0110	0xD6
		هر دو حالت	0b0101 0111	0x57
	هدر با بایت های آدرس	بایت فرمت	0b1101 1001	0xD9
		بای طول داده	0b1101 1010	0xDA
		هر دو حالت	0b0101 1011	0x5B
		بایت فرمت	0b0101 1101	0x5D
		بای طول داده	0b0101 1110	0x5E
		هر دو حالت	0b1101 1111	0xDF
		هر دو حالت: هدر با بایت- های آدرس و بدون بایت- های آدرس		
زمان بندی متعارف	هدر یک بایتی	بایت فرمت	0b1110 0101	0xE5
		بای طول داده	0b1110 0110	0xE6
		هر دو حالت	0b0110 0111	0x67
	هدر با بایت های آدرس	بایت فرمت	0b1110 1001	0xE9
		بای طول داده	0b1110 1010	0xEA
		هر دو حالت	0b0110 1011	0x6B
		بایت فرمت	0b0110 1101	0x6D
		بای طول داده	0b0110 1110	0x6E
		هر دو حالت	0b1110 1111	0xEF
		هر دو حالت: هدر با بایت- های آدرس و بدون بایت- های آدرس		

8- شروع کند

برای شروع کند، بهتر است تنظیمات پایه TX را به GPIO تغییر دهیم؛ زیرا ممکن است که UART قادر به کارکردن با سرعت 5 نباشد بنابراین بهتر است از پایه GPIO برای ارسال استفاده کرد (شکل 11). برای شروع کردن ارتباط، ابتدا باید یک آدرس مثلا 0x33 با سرعت 5 bps ارسال شود. واحد مربوطه بعد از تایید آدرس سه بایت را مثلا 0x55 0xEF 0x8F با سرعت 10.4 kbps ارسال می کند بایت اول همیشه باید 0x55 باشد. دو بایت

بعدی بایت‌های کلیدی (KB1 و KB2) نامیده می‌شوند. اگر هر دو بایت 08 08 یا 94 94 باشند پروتکل ISO-9141 است در غیر این صورت KWP2000 خواهد بود. بعد از دریافت این سه بایت، باید معکوس بایت KB2 (حدودا بین 25 تا 50 میلی ثانیه بعد از دریافت) به واحد مربوطه ارسال شود. را حدودا بین 50 تا 300 میلی ثانیه بعد ارسال می‌کند. در این مثال (معکوس 0x8F) برابر با 0x70 است. در نهایت واحد مربوطه یک بایت را معکوس آدرس آن واحد است را ارسال می‌کند که از آن برای درخواست‌ها استفاده می‌شود. در اینجا شروع ارتباط با موفقیت پایان پذیرفته و واحد مربوطه آماده پاسخ به درخواست‌ها است. به طور کلی این مراحل حدودا 2 ثانیه طول خواهد کشید.



شکل 28- زمان بندی شروع کند در KLine.

9- انواع درخواست‌ها به ECU

در جدول 11 همه انواع درخواست‌ها در KLine لیست شده است. هر کدام از درخواست‌ها در KLine توسط یک بایت مشخص می‌شوند.

جدول 11- همه انواع درخواست‌ها در KLine.

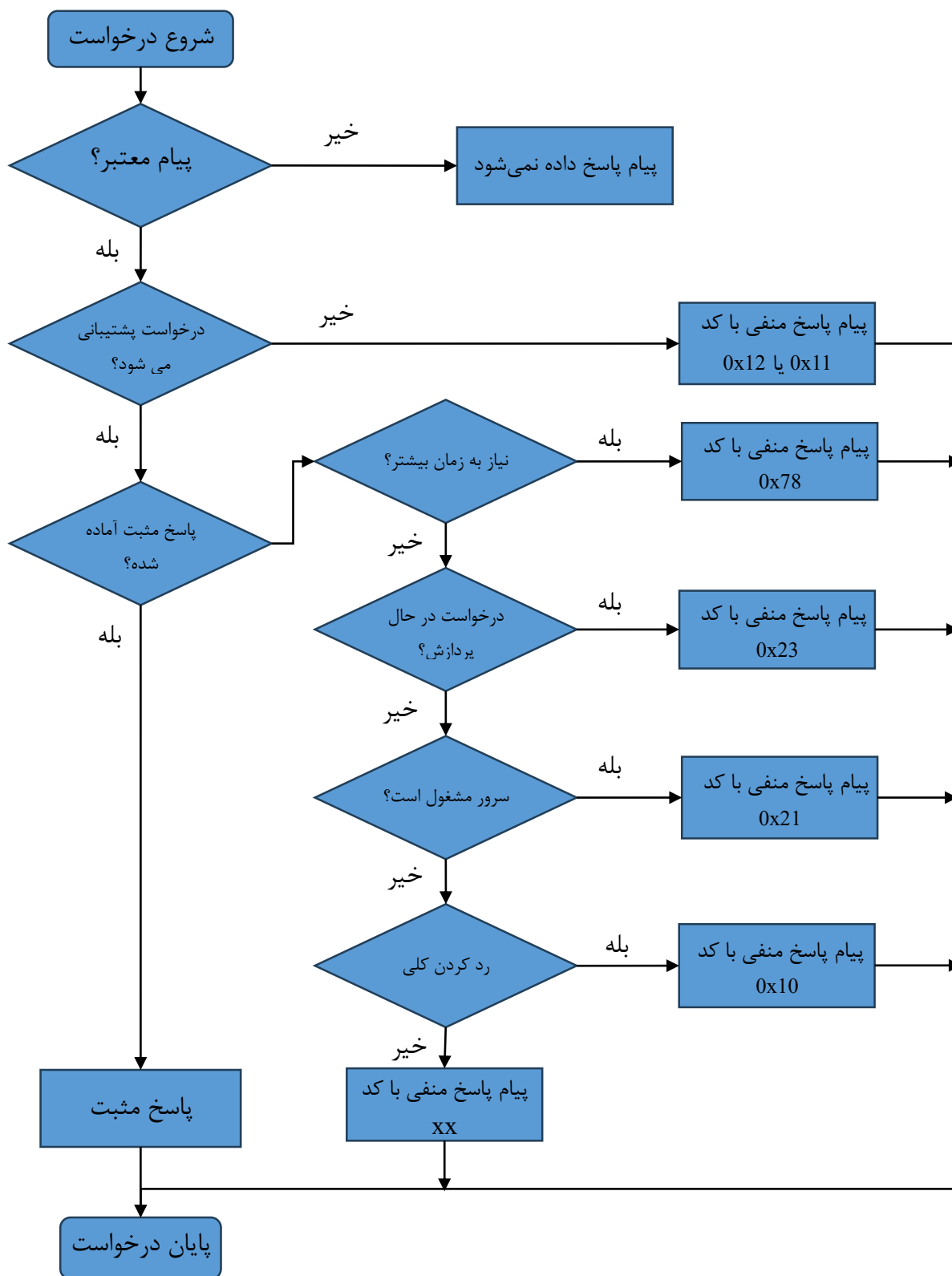
اسم درخواست	کد درخواست hex	اسم درخواست	کد درخواست hex
شروع سشن عیب‌یابی	10	کنترل عملگرها به وسیله شناسه عمومی	2F
ریست ECU	11	کنترل عملگرها به وسیله شناسه محلی	30
خواندن داده‌های فریز شده	12	شروع روتین به وسیله شناسه محلی	31
خواندن کدهای خطا	13	پایان روتین به وسیله شناسه محلی	32
پاک کردن خطاها	14	درخواست نتیجه روتین به وسیله شناسه محلی	33
خواندن وضعیت کدهای خطا	17	درخواست دانلود	34

35	درخواست آپلود	18	خواندن کدهای خطا با در نظر گرفتن نوع خطا مدنظر
36	انتقال داده	1A	خواندن اطلاعات ECU
37	درخواست پایان انتقال	20	پایان سشن عیب یابی
38	شروع روتین به وسیله آدرس	21	خواندن داده به وسیله شناسه محلی
39	پایان روتین به وسیله آدرس	22	خواندن داده به وسیله شناسه عمومی
3A	درخواست نتیجه روتین به وسیله آدرس	23	خواندن داده های حافظه با دادن آدرس
3B	نوشتن داده به وسیله شناسه محلی	26	تنظیم نرخ داده
3D	نوشتن داده به وسیله آدرس	27	مجوز دسترسی
3E	حضور درخواست کننده	2C	تعریف شناسه محلی به صورت پویا
		2E	نوشتن در حافظه به وسیله شناسه عمومی

1-9- الگوریتم پاسخ دهی ECU به درخواست ارسال شده به آن

در شکل 29 الگوریتم پاسخ دهی ECU به درخواست ارسال شده به آن ترسیم شده است. در صورتی که پیام ارسال شده معتبر نباشد به آن هیچ پاسخی نمی دهد. در صورتی که درخواست یا زیر درخواست ارسال شده پشتیبانی نشود با کد خطای 0x11 یا 0x12 پاسخ منفی به درخواست ارسال می کند. در صورتی که پاسخ مثبت آماده شده باشد آنرا ارسال می کند در غیر این صورت ابتدا بررسی می کند که آیا به زمان بیشتری نیاز دارد یا نه. اگر زمان بیشتری نیاز دارد با ارسال پاسخ منفی با کد 0x78 به درخواست کننده، آن را مطلع می کند. در مرحله بعدی بررسی می کند اگر روتین هنوز تمام نشده باشد آنگاه خطای 0x23 را ارسال می کند. در صورتی که سرور مشغول باشد و نتواند به درخواست پاسخ دهد با ارسال کد 0x21 به درخواست کننده اطلاع می دهد که سرور مشغول است و درخواست را مجدداً با تاخیر تکرار کند. و در نهایت یا کد خطای رد کردن کلی 0x10 را می فرستد یا خطای متناظر با نوع درخواست ارسال شده را.

ecu.sis00g.com



شکل 29- الگوریتم پاسخ دهی به درخواست در KLine.

9-2- درخواست شروع ارتباط

این درخواست در قسمت قبل بررسی شد.

9-3- درخواست پایان ارتباط

زمانیکه لازم باشد تا ارتباط ایجاد شده را ببندیم می‌توانیم درخواست پایان ارتباط با کد درخواست 0x82 را ارسال کنیم. البته دقت شود که اگر از آخرین پاسخ ECU مدت زمان P_{3max} سپری شده باشد و در این مدت زمان هیچ درخواست دیگری ارسال نشود، به صورت خودکار ارتباط توسط ECU بسته می‌شود. ساختار پیام درخواست، پاسخ مثبت و پاسخ منفی به ترتیب در شکل 30، شکل 31 و شکل 32 نشان داده شده‌اند.

xx	82	xx	xx	81 or C1
چک سام	سرویس درخواست پایان ارتباط	بایت آدرس منبع	بایت آدرس هدف	بایت فرمت آدرس دهی فیزیکی : 81 آدرس دهی گروهی: C1

شکل 30 - ساختار پیام درخواست پایان ارتباط در KLine.

xx	C2	xx	xx	xx
چک سام	پاسخ مثبت به سرویس درخواست پایان ارتباط	بایت آدرس منبع	بایت آدرس هدف	بایت فرمت

شکل 31 - ساختار پیام جواب مثبت به درخواست پایان ارتباط در KLine.

xx	10	82	7F	xx	xx	xx	xx
چک سام	رد درخواست	پایان ارتباط سرویس درخواست	سرویس درخواستی منفی به	بایت طول پیام	بایت آدرس منبع	بایت آدرس منبع	بایت فرمت

شکل 32- ساختار پیام جواب منفی به درخواست پایان ارتباط در KLine.

4-9- درخواست دسترسی به پارامترهای زمان بندی

این درخواست پارامترهای زمان بندی در ارتباط KLine را برمی گرداند. کد درخواست آن 0x83 بوده و داری یک زیرکد هم است که می تواند 0x00، 0x01، 0x02 و یا 0x03 باشد (شکل 33 الف)). کد 0x00 محدودیت زمان بندی را برمی گرداند 0x01 پارامترهای زمان بندی با مقادیر قرار دادی تنظیم می کند. برای خواندن مقادیر پارامترهای زمان بندی فعلی باید 0x02 را ارسال کنیم و برای تنظیم مقادیر پارامترهای زمان بندی باید از کد 0x03 استفاده کنیم. برای تنظیم پارامترها باید از فرمت نشان داده شده در شکل 25 (ب) استفاده کرد. دقت شود در صورتی هدف تنظیم پارامتر باشد (یعنی 0x03)، در ساختار ارسال پیام 5 بایت بعدی ارسال می شود.

XX	XX	XX	83	XX	XX	XX
بایت فرمت	بایت آدرس منبع	بایت آدرس منبع	سرویس درخواست زمان بندی	نوع درخواست خواندن محدودیت: 0x00 تنظیم به مقادیر قراردادی: 0x01 خواندن مقادیر فعلی: 0x02 تنظیم زمان بندی: 0x03	مقدار تنظیمات (5 بایت، فقط در هنگام ارسال کد 0x03)	یک بایت

P _{2min}	P _{2max}	P _{3min}	P _{3max}	P _{4min}
-------------------	-------------------	-------------------	-------------------	-------------------

ضریب (ms): 0.5 25 0.5 250 0.5

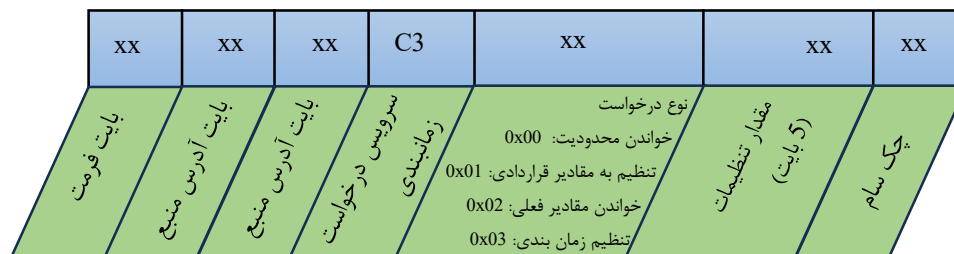
(الف)

(ب)

شکل 33 - الف) ساختار پیام درخواست دسترسی به زمان بندی در KLine (ب) فرمت ارسال پارامترهای تنظیم

زمان بندی.

جواب درخواست اگر پاسخ مثبت باشد به صورت شکل 34 خواهد بود. فرمت پارامترهای تنظیمات، همان فرمت ارسال نشان داده شده در شکل 33 (ب) است. برای محاسبه مقدار زمان متناسب با هر پارامتر کافی است که بایت مورد نظر را در ضریب متناسب با آن ضرب کرد. به عنوان مثال مثلاً برای P_{2min} باید بایت اول را در عدد 0.5 ضرب کنیم.



شکل 34- ساختار پیام جواب مثبت به درخواست دسترسی به زمان بندی در KLine.

در صورتی که جواب منفی باشد آنگاه پاسخ به شکل 35 خواهد بود.



شکل 35 - ساختار پیام جواب منفی به درخواست دسترسی به زمان بندی در KLine.

به عنوان مثال برای دسترسی به محدودیت زمان بندی ECU از نوع SIEMENS باید کد {0x82,0x11,0xF1,0x83,0x00,0x07} را ارسال کنیم. در پاسخ جواب به صورت {0x87,0xF1,0x11,0xC3,0x00,0x00,0xFE,0x01,0x28,0x00,0x73} ارسال خواهد شد؛ بنابراین داریم:

$$P_{2min} = 0x00 * 0.5 \text{ ms} = 0 \text{ ms}$$

$$P_{2max} = 0xFE * 25 \text{ ms} = 6350 \text{ ms}$$

$$P_{3min} = 0x01 * 0.5 \text{ ms} = 0.5 \text{ ms}$$

$$P_{3max} = 0x28 * 250 \text{ ms} = 10000 \text{ ms}$$

$$P_{4min} = 0x00 * 0.5 \text{ ms} = 0 \text{ ms}.$$

9-5- درخواست اعلان حضور

برای جلوگیری از بسته شدن پروتکل ارتباطی پیش زمان P_{3max} می‌توان از کد درخواست 0x3E استفاده کرد (شکل 36). جواب مثبت و منفی این درخواست به ترتیب در شکل 37 و شکل 38 نشان داده شده است.

xx	xx	xx	3E	xx	xx
بیت فرمت	بیت آدرس هدف	بیت آدرس منبع	سرویس زنده نگه داشتن ارتباط	وجود پاسخ پاسخ: 0x01 عدم پاسخ: 0x02	یک سام

شکل 36- ساختار پیام اعلان حضور در KLine.

xx	xx	xx	7E	xx
بیت فرمت	بیت آدرس هدف	بیت آدرس منبع	پاسخ مثبت به سرویس زنده نگه داشتن ارتباط	یک سام

شکل 37- ساختار پیام جواب مثبت به درخواست اعلان حضور در KLine.

xx	xx	xx	7F	3E	xx	xx
بیت فرمت	بیت آدرس هدف	بیت آدرس منبع	پاسخ منفی به سرویس درخواستی	زنده نگه داشتن ارتباط درخواست	رد شدن پیام کد علت	یک سام

شکل 38- ساختار پیام پاسخ منفی به درخواست اعلان حضور در KLine.

9-6- درخواست ریست واحد الکترونیکی

می‌توان کل واحد الکترونیکی یا قسمتی از آن را ریست کرد. برای ریست کردن ECU می‌توان از دستور 0x11 استفاده کرد)

شکل 39). پاسخ مثبت و منفی به این دستور به ترتیب در

شکل 41 آورده شده است.

40 و

XX	XX	XX	11	XX	XX
بایت فرمت	بایت آدرس هدف	بایت آدرس منبع	سرویس ریست	نوع ریست ریست بدون حفظ ارتباط: 0x01 ریست با حفظ ارتباط: 0x02	چک سام

شکل 39 - ساختار پیام درخواست ریست در KLine.

XX	XX	XX	51	XX	XX
بایت فرمت	بایت آدرس هدف	بایت آدرس منبع	پاسخ مثبت به سرویس ریست	نوع ریست ریست بدون حفظ ارتباط: 0x01 ریست با حفظ ارتباط: 0x02	چک سام

شکل 40-ساختار پیام پاسخ مثبت به درخواست ریست در KLine.

XX	XX	XX	7F	XX	XX	XX
بایت فرمت	بایت آدرس هدف	بایت آدرس منبع	پاسخ منفی به سرویس ریست	سرویس ریست	کد علت رد درخواست	چک سام

شکل 41- ساختار پیام جواب منفی به درخواست ریست در KLine.

9-7- درخواست خواندن اطلاعات واحد الکترونیکی

برای خواندن اطلاعات واحد الکترونیکی از درخواست 0x1A استفاده می‌شود (شکل 42). این اطلاعات شامل VIN خودرو، نسخه نرم افزار واحد الکترونیکی، تاریخ برنامه ریزی و... می‌شود.

XX	XX	XX	1A	XX	XX
بایت فرمت	بایت آدرس هدف	بایت آدرس منبع	سرویس خواندن مشخصات ECU	نوع زیر سرویس درخواستی	چک سام

شکل 42- ساختار پیام درخواست دسترسی به مشخصات در KLine.

XX	XX	XX	5A	XX	XX
بایت فرمت	بایت آدرس هدف	بایت آدرس منبع	سرویس ریست پاسخ مثبت به	داده ها	چک سام

شکل 43- ساختار پیام جواب مثبت به درخواست دسترسی به مشخصات در KLine.

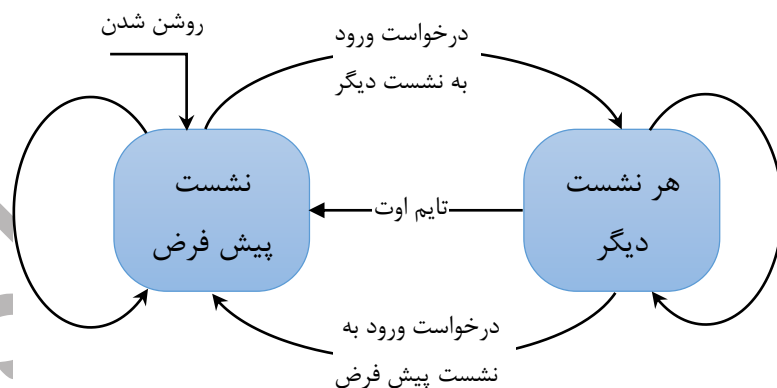
XX	XX	XX	7F	1A	XX	XX
بایت فرمت	بایت آدرس هدف	بایت آدرس منبع	سرویس پاسخ منفی به	سرویس خواندن مشخصات ECU	رد درخواست کد علت	چک سام

شکل 44- ساختار پیام جواب منفی به درخواست دسترسی به مشخصات در KLine.

8-9- سرویس درخواست شروع نشست

این سرویس انواع متفاوتی از نشست‌های عیب‌یابی را فعال می‌کند. به‌منظور اجرای هر سرویس عیب‌یابی باید ابتدا نشست متناظر با آن شروع شود. نکات این سرویس در زیر لیست شده است (شکل 45).

- در هر زمان فقط یک نشست عیب‌یابی می‌تواند فعال باشد.
- اگر هیچ نشست عیب‌یابی درخواست داده نشود، به‌صورت خودکار واحد الکترونیکی در حالت نشست نرمال/ همان قراردادی قرار می‌گیرد.
- بعد از پایان زمان تایم اوت هر نشست، واحد الکترونیکی به‌صورت خودکار به حالت نشست قراردادی/نرمال برمی‌گردد.
- اگر واحد الکترونیکی قادر نباشد به سرویس نشست درخواستی سوئیچ کند در نشست جاری خود باقی می‌ماند.
- اگر واحد الکترونیکی به‌صورت فانکشنالی درخواست نشست عیب‌یابی دریافت کند به آن هیچ پاسخی نخواهد داد.



شکل 45- نمودار حالت نشست‌های عیب‌یابی.



شکل 46- ساختار پیام درخواست شروع سشن در KLine.

XX	XX	XX	50	XX	XX
بایت فرمت	بایت آدرس هدف	بایت آدرس منبع	پاسخ مثبت به سرویس ریست	داده ها	چک سام

شکل 47- ساختار پیام جواب مثبت به درخواست شروع سشن در KLine.

XX	XX	XX	7F	10	XX	XX
بایت فرمت	بایت آدرس هدف	بایت آدرس منبع	پاسخ منفی به سرویس	سرویس خواندن مشخصات ECU	رد درخواست	چک سام

شکل 48 - ساختار پیام جواب منفی به درخواست شروع سشن در KLine.

9-9- درخواست پایان نشست

این سرویس درخواست غیرفعال کردن نشست فعلی و بازگشت به نشست پیش فرض را فراهم می کند (شکل 49). ساختار پیام پاسخ مثبت یا منفی به این سرویس به ترتیب در شکل 50 و

شکل 51 نشان داده شده است. برای استفاده از این درخواست باید نکات زیر را در نظر گرفت

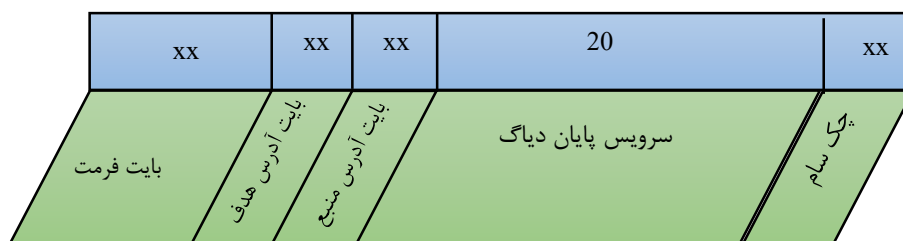
برای استفاده از این سرویس باید ارتباط ایجاد شده باشد و نشست باید در حال اجرا باشد که بتوان آن را پایان داد. در غیر این صورت اگر نشست فعال نباشد، جواب درخواست منفی خواهد بود

این درخواست را نمی توان برای غیرفعال کردن نشست پیش فرض استفاده کرد.

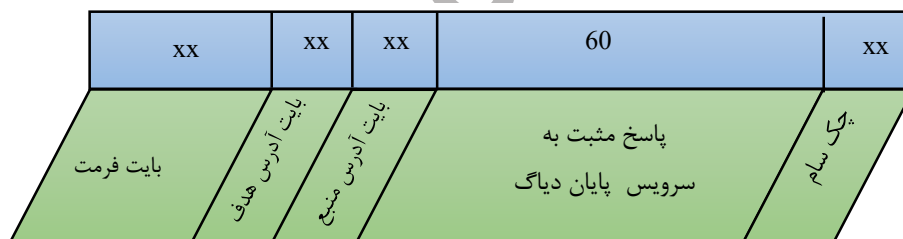
هنگامی که پاسخ مثبت ارسال می شود تنظیمات زمان بندی به تنظیمات زمان بندی مرتبط در نشست پیش فرض برگردانده می شود.

با ارسال پاسخ مثبت، در صورتی که در نشست فعال شده مجوز دسترسی خاصی امکان پذیر شده باشد با پایان یافتن نشست مجوز دسترسی لغو خواهد شد.

در صورتی که پاسخ منفی به پایان نشست ارسال شود، در نشست فعلی باقی می ماند.



شکل 49- ساختار پیام درخواست پایان سشن در KLine.



شکل 50- ساختار پیام جواب مثبت به درخواست پایان سشن در KLine.



شکل 51 - ساختار پیام جواب منفی به درخواست پایان سشن در KLine.

9-10- درخواست خواندن خطا

9-10-1- کد خطا 18

به وسیله این سرویس می توان همه یا گروهی از کدهای خطا و وضعیت آنها را از حافظه خواند. همان طور که در شکل 52 نشان داده شده اگر مقدار بایت وضعیت DTC برابر با 0x00 آنگاه کدهای خطای شناسایی شده را به همراه وضعیت آنها (در صورت وجود) ارسال می شود. سایر مقادیر را می توان به تناسب استفاده کرد. می توان همه کدهای خطا را خواند (FF00) یا گروهی از کدها خطا را خواند (0000 برای کدهای پیشرانه یا همان قوای محرکه، 4000 کدهای خطای شاسی، 8000 کدهای خطای بدنه و C000 مدهای خطای مربوط به شبکه و سیم کشی).

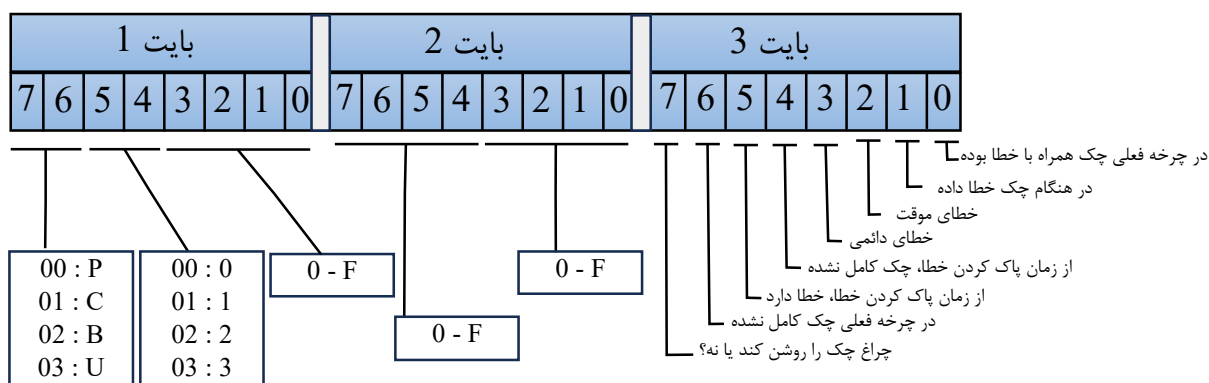
XX	XX	XX	18	XX	XXXX	XX
بیت 7	بیت 6	بیت 5	بیت 4	بایت وضعیت DTC: درخواست DTC های ... 00: شناسایی شده و وضعیت آنها 01: پشتیبانی شده و وضعیت آنها 02: (دو بایتی) شناسایی شده و وضعیت آنها 03: (دوبایتی) شناسایی شده و وضعیت آنها 04: آخر	0000 : همه کدهای خطای پیشرانه 4000 : همه کدهای خطای شاسی 8000 : همه کدهای خطای بدنه C000 : همه کدهای خطای شبکه FF00 : همه کدهای خطا	بیت 3

شکل 52 - ساختار پیام درخواست خواندن خطاها در KLine.

در جواب این درخواست همیشه تعداد خطاها، کد خطا و وضعیت خطا وجود خواهد داشت (شکل 53). اگر تعداد خطاها صفر باشد در پاسخ تعداد خطاها صفر ذکر شده و بقیه داده ها ارسال نخواهد شد. در صورتی که کد خطا(هایی) وجود داشته باشد به ازای هر کد خطا سه بایت در نظر گرفته شده است. روش استخراج کد خطا در شکل 54 نشان داده شده است. دو بیت بارزش بایت اول (7 و 6 ام) گروه خطا را مشخص می کنند اگر به ترتیب برابر با 0، 1، 2 یا 3 باشند آنگاه گروه خطا P، C، B و یا U خواهند بود. دو بیت بعدی بایت اول (5 و 4 ام) مشخص می کند که کد خطا عمومی است یا یک کد اختصاصی برای خودرو ساز است. 4 بیت کم ارزش بایت اول (0 تا 3 ام) رقم سوم کد خطا را مشخص می کند و دو رقم دیگر هم از بایت دوم حاصل می شود. بایت سوم اطلاعاتی درباره کد خطا می دهد. مثلاً اگر بیت 7 ام آن یک باشد این به معنای آن است که این کد خطا موجب روشن شدن چراخ چک خواهد شد. همچنین در صورتی که کد خطا، یک کد خطای دائمی باشد بیت 3 ام یک می شود و در صورتی که خطا موقت باشد، بیت 2 ام یک می شود.

XX	XX	XX	XX	58	XX	XXXXXX	...	XXXXXX	XX
بایت فرمت	بایت آدرس هدف	بایت آدرس منبع	بایت طول (در صورت لزوم)	پاسخ مثبت به سرویس	تعداد کدهای خطا	اولین کد خطا		آخرین کد خطا	نام

شکل 53- ساختار پیام جواب مثبت به درخواست خواندن خطا در KLine



شکل 54- روش ساخت کد خطا با توجه به داده ارسالی از واحد الکترونیکی.

در صورتی که واحد کنترل الکترونیکی به هر دلیل نتواند به سرویس خواندن خطا جواب مثبت دهد به صورت نشان داده شده در شکل 55 به آن جواب خواهد داد.

XX	XX	XX	7F	18	XX	XX
بایت فرمت	بایت آدرس هدف	بایت آدرس منبع	پاسخ منفی به سرویس	سرویس پایان یابی	کد علت رد درخواست	چک سام

شکل 55 - ساختار پیام جواب منفی به سرویس درخواست خواندن خطا در KLine.

2-10-9- کد خطا 13

به وسیله این سرویس می توان همه یا گروهی از کدهای خطا را از حافظه خواند. ساختار پیام این سرویس در شکل 56 آورده شده است. این سرویس وضعیت کد خطا را بر نمی گرداند و فقط کد خطا را ارسال می کند. بنابراین کدهای خطای ارسالی دو بایتی هستند (شکل 57). در صورت جواب منفی پاسخ به صورت ... خواهد بود.

XX	XXXX	13	XX	XX	XX
بایت فرمت	0000 : همه کدهای خطای پیشرفته 4000 : همه کدهای خطای شناسی 8000 : همه کدهای خطای بدنه C000 : همه کدهای خطای شبکه FF00 : همه کدهای خطا	سرویس درخواست خواندن خطا	بایت آدرس منبع	بایت آدرس هدف	بایت فرمت

شکل 56- ساختار پیام درخواست خواندن خطاها در KLine.

XX	XXXX	...	XXXX	XX	53	XX	XX	XX	XX
بایت فرمت	اولین کد خطا		آخرین کد خطا		تعداد کدهای خطا	پاسخ مثبت به سرویس	بایت طول (در صورت لزوم)	بایت آدرس منبع	بایت آدرس هدف

شکل 57- ساختار پیام جواب مثبت به درخواست خواندن خطا در KLine.

XX	XX	XX	7F	13	XX	XX
بایت فرمت	بایت آدرس هدف	بایت آدرس منبع	پاسخ منفی به سرویس	سرویس پایان یابی	کد علت رد درخواست	بایت فرمت

شکل 58- ساختار پیام جواب منفی به سرویس درخواست خواندن خطا در KLine.

3-10-9- کد خطا 17

این سرویس هم برای خواندن کد خطا مورد استفاده قرار می‌گیرد و مشابه کد خطای 13 است با این تفاوت که بایت وضعیت کد خطا را همراه با کد خطا ارسال می‌کند یعنی کدهای خطای ارسالی سه بایتی هستند.

XX	XXXX	13	XX	XX	XX
بایت فرمت	سرویس درخواست خواندن خطا	بایت آدرس منبع	بایت آدرس هدف	بایت فرمت	بایت فرمت
0000 - 3FFF : کدهای خطای پیش‌رانه 4000 - 7FFF : کدهای خطای شاسی 8000 - BFFF : کدهای خطای بدنه C000 - FEFF : کدهای خطای شبکه FF00 : همه کدهای خطا					

شکل 59- ساختار پیام درخواست خواندن خطاها در KLine.

XX	XXXXXX	...	XXXXXX	XX	57	XX	XX	XX	XX
بایت فرمت	اولین کد خطا		آخرین کد خطا	تعداد کدهای خطا	پاسخ مثبت به سرویس	بایت طول (در صورت لزوم)	بایت آدرس منبع	بایت آدرس هدف	بایت فرمت

شکل 60- ساختار پیام جواب مثبت به درخواست خواندن خطا در KLine

XX	XX	XX	7F	17	XX	XX
بایت فرمت	بایت آدرس هدف	بایت آدرس منبع	پاسخ منفی به سرویس	سرویس پایان یابی	کد علت رد درخواست	بایت فرمت

شکل 61- ساختار پیام جواب منفی به سرویس درخواست خواندن خطا در KLine.

9-11- درخواست پاک کردن خطا

این سرویس برای پاک کردن کد خطاهای عیب‌یابی و داده‌های ذخیره شده متناظر با آن در حافظه، استفاده می‌شود. ساختار پیام این درخواست در

شکل 62 نشان داده شده است، به طور کلی خطاها را می‌توان به چهار دسته پیش‌ران، بدنه، شاسی و شبکه تقسیم می‌شوند. پاک کردن خطاها می‌تواند به صورت گرهی یا تکی صورت بپذیرد. اگر کد ارسالی برابر با یکی از مقادیر 0x0000، 0x4000، 0x8000، 0xC000 و یا 0xFF00 باشد، گروهی از خطاها پاک می‌شود در غیر این صورت خطاها به صورت تکی پاک می‌شوند.

XX	XX	XX	14	XXXX	XX
بایت فرمت	بایت آدرس هدف	بایت آدرس منبع	پاک کردن خطا سرویس درخواست	0000 : همه کدهای خطای پیش‌ران 0001 – 3FFF : کد خطای خاص پیش‌ران 4000 : همه کدهای خطای شاسی 4001 – 7FFF : کد خطای خاص پیش‌ران 8000 : همه کدهای خطای شاسی 8001 – BFFF : کد خطای خاص پیش‌ران C000 : همه کدهای خطای شاسی C001 – FEFF : کد خطای خاص پیش‌ران FF00 : همه کدهای خطا	بایت کد خطا

شکل 62- ساختار پیام درخواست پاک کردن خطاها در KLine.

XX	XX	XX	54	XXXX	XX
بایت فرمت	بایت آدرس هدف	بایت آدرس منبع	پاسخ مثبت به سرویس پایان دیاگ	ارسال شده DTC	بایت کد خطا

شکل 63 - ساختار پیام جواب مثبت به درخواست پاک کردن خطاها در KLine.

XX	XX	XX	7F	14	XX	XX
بایت فرمت	بایت آدرس هدف	بایت آدرس منبع	پاسخ منفی به سرویس درخواست پاک کردن خطا	پاک کردن خطا سرویس درخواست	رد درخواست کد علت	بایت کد خطا

شکل 64 - ساختار پیام جواب منفی به درخواست پاک کردن خطاها در KLine.

به عنوان مثال برای پاک کردن خطاهای پیشرانه در SIEMENS باید درخواست با متن پیام {0x83,0x11,0xF1,0x14,0x00,0x00,0x99} ارسال شود. یا برای پاک کردن خطای P0120 متن پیام درخواست به صورت {0x83,0x11,0xF1,0x14,0x01,0x20,0xBA} خواهد بود.

9-12- سرویس خواندن داده

9-12-1- سرویس خواندن داده با شناسه محلی

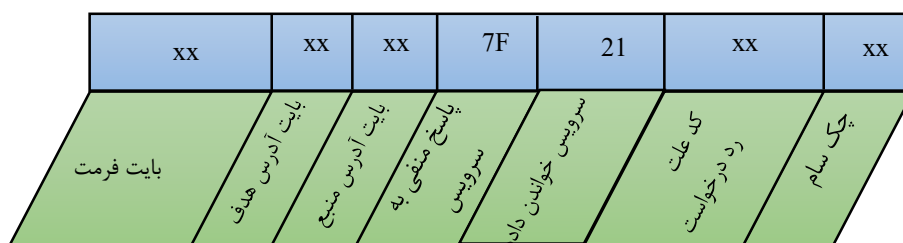
این سرویس درخواست خواندن داده با شناسه محلی از واحد کنترل الکترونیکی را ارسال می کند. منظور از شناسه محلی خواندن داده از حافظه فلش یا RAM است (شکل 65). در پاسخ به این درخواست، یک پیام با ساختار نشان داده شده در شکل 66 ارسال می شود. ساختار متن پیام بسته شرکت سازنده دارد.



شکل 65- ساختار پیام خواندن داده با شناسه محلی حضور در KLine.



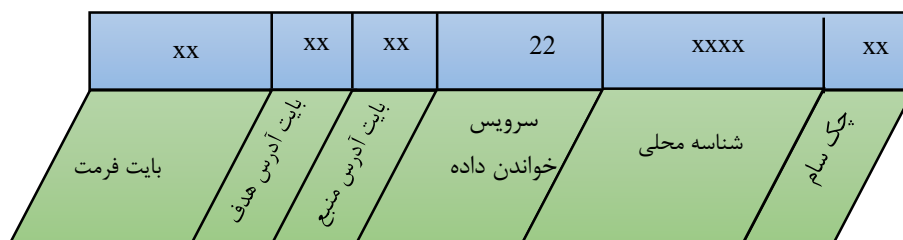
شکل 66 - ساختار پیام جواب مثبت به درخواست دسترسی به مشخصات در KLine.



شکل 67- ساختار پیام جواب منفی به درخواست خواندن داده در KLine.

2-12-9- سرویس خواندن داده با شناسه عمومی

این سرویس مشابه سرویس قبلی است با این تفاوت که از شناسه عمومی به جای شناسه محلی استفاده می کند.



شکل 68- ساختار پیام خواندن داده با شناسه محلی حضور در KLine.



شکل 69- ساختار پیام جواب مثبت به درخواست دسترسی به مشخصات در KLine.



شکل 70- ساختار پیام جواب منفی به درخواست خواندن داده در KLine.

9-13- سرویس کنترل ورودی/خروجی (تست عملگرها)

9-13-1- سرویس کنترل ورودی/خروجی با شناسه محلی

این سرویس برای تست عملگرها (مانند انژکتورها، رله پمپبنزین، استپر موتور و...) مورد استفاده قرار می گیرد (شکل 71). واحد کنترل الکترونیکی بعد از دریافت درخواست، حالت عملگر مربوطه را به حالت مورد نظر تغییر می دهد (شکل 72)

XX	XX	XX	30	XX	XXXX	XX
بایت فرمت	بایت آدرس هدف	بایت آدرس منبع	سرویس تست عملگر	شناسه عملگر	بایت های کنترلی	بک سام

شکل 71- ساختار پیام درخواست تست عملگر در KLine.

XX	XX	XX	70	XX	XXXX	XX
بایت فرمت	بایت آدرس هدف	بایت آدرس منبع	پاسخ مثبت به تست عملگر	شناسه عملگر	بایت های کنترلی	بک سام

شکل 72- ساختار پیام جواب مثبت به درخواست تست عملگر در KLine.

XX	XX	XX	7F	30	XX	XX
بایت فرمت	بایت آدرس هدف	بایت آدرس منبع	پاسخ منفی به سرویس	سرویس درخواست تست عملگر	رد علت درخواست	بک سام

شکل 73- ساختار پیام جواب منفی به درخواست تست عملگر در KLine.

بایت های کنترلی مشخص می کنند که چگونه عملگرها تست و کنترل شوند. حالت های مختلف بایت اول در ادامه شرح داده شده است:

- برگرداندن کنترل به واحد الکترونیکی 0x00: این مقدار به واحد الکترونیکی اطلاع می دهد که دیگر دستگاه درخواست کننده هیچ کنترلی روی ورودی یا خروجی ندارد.

- گزارش وضعیت فعلی 0x01: این مقدار از واحد الکترونیکی درخواست می‌کند که وضعیت فعلی سیگنال ورودی/داخلی/خروجی را به اطلاع دستگاه عیب برساند.
- ریست به حالت پیش‌فرض 0x04: این مقدار از واحد الکترونیکی درخواست می‌کند که مقدار فعلی سیگنال ورودی/داخلی/خروجی را به مقدار پیش‌فرض آن بازگرداند.
- نگه‌داشتن مقدار فعلی 0x05: این مقدار از واحد الکترونیکی درخواست می‌کند که مقدار فعلی سیگنال ورودی/داخلی/خروجی را نگه دارد.
- تنظیم کوتاه‌مدت 0x07: این مقدار به واحد الکترونیکی دستور می‌دهد که مقدار فعلی سیگنال ورودی/داخلی/خارجی را به مقدار آورده شده در متن پیام دستور تغییر دهد. با برگشت به نشت قراردادی یا برگرداندن کنترل به واحد الکترونیکی، واحد الکترونیکی مجدد کنترل را در دست می‌گیرد.
- تنظیم بلندمدت 0x08: بایت کنترلی دوم معمولاً حاوی حالت مدنظر برای تست عملگر است. مثلاً اگر 0xFF باشد به معنای آن است که مثلاً رله پمپ‌بنزین را روشن کن و اگر 0x00 باشد رله پمپ‌بنزین را خاموش کن.

2-13-9- سرویس کنترل ورودی/خروجی با شناسه عمومی

این سرویس به جای شناسه محلی از شناسه عمومی برای تست عملگر استفاده می‌کند ساختار پیام درخواست و جواب مثبت و منفی به ترتیب در شکل 74 و شکل 75 و شکل 76 ترسیم شده‌اند.

XX	XXXX	XXXX	2F	XX	XX	XX
بایت فرمت	بایت‌های کنترلی	شناسه عملگر	سرویس تست عملگر	بایت آدرس منبع	بایت آدرس هدف	بایت فرمت

شکل 74- ساختار پیام درخواست تست عملگر در KLine.

XX	XXXX	XXXX	6F	XX	XX	XX
بایت فرمت	بایت‌های کنترلی	شناسه عملگر	پاسخ مثبت به تست عملگر	بایت آدرس منبع	بایت آدرس هدف	بایت فرمت

شکل 75- ساختار پیام جواب مثبت به درخواست تست عملگر در KLine.

xx	xx	xx	7F	2F	xx	xx
بایت فرمت	بایت آدرس هدف	بایت آدرس منبع	پاسخ منفی به سرویس	سرویس درخواست تست عملگر	رد درخواست کد علت	چک سام

شکل 76- ساختار پیام جواب منفی به درخواست تست عملگر در KLine.

3-13-9- رویه تست عملگرها

رویه تست عملگرها معمولاً بدین ترتیب است که ابتدا حالت فعلی عملگر را نگه داشته سپس اختیار از واحد الکترونیکی به دستگاه درخواست کننده محول شده و تست عملگر انجام می شود و در نهایت کنترل عملگر مجدد به واحد کنترل الکترونیکی واگذار می شود.

10- هندل کردن خطا

1-10-1- هندل کردن خطا در هنگام شروع ارتباط

اگر درخواست کننده در هنگام سرویس درخواست شروع ارتباط خطایی (زمان بندی، داده و یا عدم پاسخ) را شناسایی کند، باید حداقل به مدت w_s صبر کند و سپس دوباره درخواست خود را ارسال کند. اگر واحد کنترل الکترونیکی خطای توالی را تشخیص دهد، بلافاصله می تواند درخواست سرویس شروع را ارسال کرد.

2-10-2- هندل کردن خطا در هنگام تبادل داده

1-10-2-1- شناسایی خطا توسط ECU در هنگام ارسال داده

ECU اعتبار پیام را توسط بایت چک سام و تعداد بایت ها بررسی می کند قبل از اینکه P_{2max} سپری شده باشد. اگر خطایی شناسایی شود هیچ پاسخی ارسال نمی کند و درخواست را نادیده می گیرد. در هنگامی که چک سام و تعداد بایت های پیام صحیح باشد؛ اما اگر خطایی در قالب بندی یا محتوای پیام وجود داشته باشد، برای اینکه درخواست کننده را مطلع کند پاسخ منفی با کد خطای متناسب با آن را ارسال می کند.

2-10-2-2- شناسایی خطا توسط درخواست کننده در پیام پاسخ

درخواست کننده هم باید پیام پاسخ را از لحاظ چک سام و تعداد بایت های دریافتی بررسی کند. در صورت تشخیص خطا یا عدم ارسال پاسخ در زمان P_{2max} ، باید دو بار دیگر درخواست را ارسال کند. اگر پاسخ صحیح دریافت نشد مشخص می شود که خطایی در ارتباط وجود دارد که باید بررسی شود.

3-2-10- ECU تشخیص می‌دهد که در پاسخی که ارسال کرده خطا دارد

باتوجه به اینکه ارسال و دریافت داده در KWP روی یک خط صورت می‌گیرد؛ بنابراین پاسخی که ECU ارسال می‌کند را خودش دریافت می‌کند؛ بنابراین اگر در پاسخ ارسال شده خطایی را شناسایی کند، ممکن است کار نکند! یا اینکه بعد از پایان فعالیت خط (حالت بیکاری) مجدد پاسخ صحیح را ارسال کند در زمان P₂.

4-2-10- درخواست‌کننده تشخیص دهد که در پیام درخواست خطا وجود دارد.

همان‌طور که بیان شد ارسال و دریافت داده در KWP روی یک خط صورت می‌گیرد؛ بنابراین درخواستی که درخواست‌کننده ارسال می‌کند را خودش دریافت می‌کند؛ بنابراین اگر در درخواست ارسال شده خطایی را شناسایی کند، بعد از پایان فعالیت خط (حالت بیکاری) شاید لازم باشد مجدد درخواست صحیح را ارسال کند در زمان P₂.