

یک واسط کاربر شی‌گرا و چندلایه برای کنترل تردد وسایط نقلیه با استفاده از RFID

فاطمه نفر^۱، حسین شمسی^۲

۱. دانشجوی کارشناسی ارشد مخابرات امن، دانشگاه علم و صنعت ایران، fatemeh.nafar@gmail.com

۲. استادیار دانشکده برق، دانشگاه صنعتی خواجه نصیرالدین طوسی

تاریخ دریافت: ۹۲/۱۲/۱۷ تاریخ پذیرش: ۹۳/۵/۱۴

چکیده

در این مقاله یک سامانه RFID فعال و امن در فرکانس ۲/۴ GHz طراحی و پیاده‌سازی شده است. این سامانه مطابق با استاندارد IEEE ۸۰۲.۱۵.۴ و با هدف کنترل بی‌سیم تردد وسایط نقلیه توسعه یافته است به طوری که قابلیت اجرای شش عملیات بی‌سیم و کارآمد را داراست. در مرکز کنترل، از طریق ارتباط سریال با خوان (RFID Reader) با PC و با استفاده از یک واسط کاربر مبتنی بر ویندوز، تردد وسایط نقلیه تا شعاع ۶۰ متری بازخوان تحت کنترل کاربر درآمده است. با فشردن یک دکمه در محیط گرافیکی واسط، در مدت ۳۱ msec شناسایی بی‌سیم ۸ وسیله نقلیه به انجام رسیده و شماره پلاک آنها به رویت کاربر می‌رسد. در توسعه این واسط، از ایده لایه‌بندی و انتزاع استفاده شده است. معماری چندلایه در قیاس با معماری یکپارچه، از مزایایی چون قابلیت گسترش و کاهش انتشار خطا برخوردار است و پیاده‌سازی با راهبرد شی‌گرا موجب تسهیل توسعه، حفظ امنیت اطلاعات و تسریع در خطایابی و به‌روزرسانی شده است.

کلیدواژه

سامانه شناسایی با استفاده از امواج رادیویی، راهبرد شی‌گرا، معماری چندلایه، واسط کاربر

مقدمه

فعال^۸ یا پلاک الکترونیکی بر روی هر وسیله نقلیه و با استقرار بازخوان در مکانهای حساس عبور و مرور، یک شبکه بی‌سیم متمرکز ایجاد می‌شود. امکان نظارت^۹ و صدور شش فرمان بی‌سیم از سوی کاربر در نرم‌افزار واسط کاربر فراهم آمده است. این واسط کاربر با استفاده از الگوی معماری چندلایه^{۱۰} و راهبرد شی‌گرا^{۱۱} توسعه یافته است و نوآوری طراحی آن در قیاس با کارهای پیشین افزودن لایه‌ای امنیتی است.

این مقاله به شرح زیر سازماندهی شده است. در بخش ۲ مروری بر واسط‌های کاربر در مقالات خواهیم داشت. در بخش ۳ وظایف، طراحی معماری و پیاده‌سازی شی‌گرای واسط کاربر مورد بررسی قرار می‌گیرد. در بخش ۴ لایه فیزیکی به همراه مختصری از طراحی سخت‌افزار بازخوان و تگ فعال شرح داده می‌شود و سرانجام در بخش ۵ نتیجه‌گیری ارائه خواهد شد.

مرور واسط‌های کاربر در مقالات

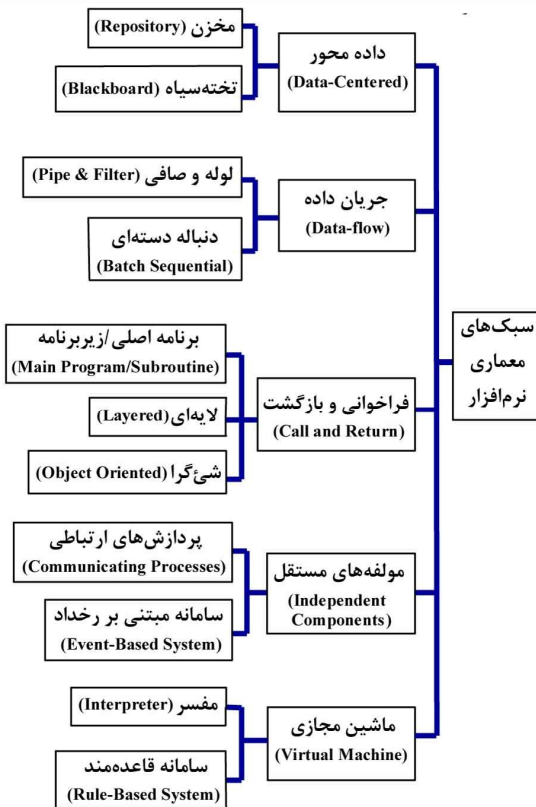
واسط کاربر گرافیکی از پنجره‌ها، دکمه‌ها^{۱۲}، جعبه‌های متنی^{۱۳}، جعبه‌های محاوره‌ای^{۱۴} و... تشکیل شده است که تبادل اطلاعات

امروزه سامانه‌های شناسایی با استفاده از امواج رادیویی^۱ از پرکاربردترین روش‌های خودکار شناسایی، ره‌گیری و سنجش از راه دور محسوب می‌شوند. این سامانه‌ها در کاربردهای متنوعی اعم از انبارداری و مدیریت کالا، کارت‌های هوشمند^۲، کنترل حمل و نقل هوشمند^۳، شناسایی ادوات جنگی و... مورد استفاده قرار می‌گیرند. سامانه‌های RFID از طریق برقراری ارتباط بی‌سیم بین یک بازخوان^۴ و یک تگ^۵ که به یک شیء متصل شده است، قادر به تبادل داده بوده و برای خواندن و نوشتن داده از امواج الکترومغناطیس بدون تماس بهره می‌گیرند. این سامانه‌ها، به سبب سرعت بالا، قابلیت اطمینان، صرفه‌جویی در هزینه و طول عمر بالا در بین سامانه‌های بی‌سیم، بسیار مورد توجه می‌باشند. در مرکز کنترل سامانه RFID، صدور فرمانهای بی‌سیم از سوی کاربر، مستلزم برقراری ارتباط بازخوان با PC و توسعه یک نرم‌افزار واسط کاربر^۶ می‌باشد [۵-۱].

در این مقاله یک سامانه RFID فعال^۷ در فرکانس ۲/۴ GHz مطابق با استاندارد IEEE ۸۰۲.۱۵.۴ و با هدف کنترل بی‌سیم تردد وسایط نقلیه طراحی و پیاده‌سازی شده است. با نصب تگ

8. Active Tag
9. Monitoring
10. Multi-Layer Architecture
11. Object Oriented
12. Buttons
13. Text boxes
14. Dialog boxes

1. Radio Frequency Identification System (RFIS)
2. Smart Card
3. Intelligent Transportation System (ITS)
4. Reader (RFID Transceiver)
5. Tag (RFID Transponder)
6. User Interface
7. Active RFID System



شکل ۱. طبقه‌بندی سبک‌های معماری نرم‌افزار Shaw & Garlan [۱۴]

در جدول ۱ مشخصات برخی سبک‌های معماری متداول و دلایل عدم انتخاب آنها در توسعه واسط کاربر پیشنهادی مورد بررسی قرار گرفته است [۱۴-۱۸].

بین کاربر نهایی^{۱۵} و سامانه را میسر می‌سازد. سهولت یادگیری و استفاده از مهمترین ویژگی‌های یک واسط کاربر کارآمد به شمار می‌رود [۸-۶]. پیچیدگی مدل واسط کاربر بر قابلیت استفاده^{۱۶} از آن، اثر نامطلوب دارد [۹]، در حالی که با ایده لایه‌بندی^{۱۷} در این واسط، از پیچیدگی‌های تحلیل و طراحی کاسته شده است.

قابلیت استفاده مجدد^{۱۸}، از ویژگی‌های کیفی مهم در طراحی یک واسط کاربر محسوب می‌شود. در رویکرد شی‌گرا، اشیاء تضمین‌کننده این قابلیت هستند. واسط کاربر مبتنی بر مدل ارائه شده در مراجع [۱۰] و [۱۱] فاقد شیء و قابلیت مزبور می‌باشد.

امتیاز طراحی واسط کاربر ارائه شده در این مقاله در قیاس با کارهای مشابه [۱۲]، افزودن لایه‌های امنیتی با هدف رمزنگاری^{۱۹} اطلاعات ارسالی و رمزگشایی^{۲۰} اطلاعات دریافتی می‌باشد تا اطلاعات از تهاجم خارجی محافظت شود.

ایجاد پایگاه داده^{۲۱} و کلاس‌های موجودیت^{۲۲} متناظر مستلزم بهره‌وری بالای CPU^{۲۳} بوده و پیامدهایی چون کاهش سرعت و اشغال فضای زیادی از حافظه را در پی دارد [۱۳]، در حالی که در این مقاله به سبب عدم نیاز به ذخیره‌سازی دائمی شماره پلاک‌ها توسط واسط کاربر، از ایجاد کلاس‌های موجودیت اجتناب شده و زمان پاسخ^{۲۴} و حجم حافظه مورد نیاز بهبود یافته است.

سبک‌های متداول معماری واسط کاربر

هر سبک معماری نرم‌افزار^{۲۵}، ساختاری متشکل از مولفه‌ها^{۲۶}، رابطه‌ها^{۲۷} و قیود^{۲۸} یکپارچه‌سازی مولفه‌ها است که نحوه سازماندهی مولفه‌ها، نحوه دست‌کاری^{۲۹} داده‌ها و نحوه برقراری ارتباط بین مولفه‌ها را مشخص می‌نماید. سبک معماری نرم‌افزار تاثیر مستقیمی بر ویژگی‌های کیفی و مخاطرات^{۳۰} بالقوه سامانه نرم‌افزاری دارد [۱۴، ۶].

در شکل ۱ طبقه‌بندی سبک‌های معماری نرم‌افزار Shaw & Garlan ارائه شده است [۱۴].

- 15 .End-User
- 16 .Usability
- 17 .Layering
- 18 .Reusability
- 19 .Encryption
- 20 .Decryption
- 21 .Data Base
- 22 .Entity Class
- 23 .Central Processing Unit(CPU)
- 24 .Response Time
- 25 .Software Architecture Style
- 26 .Component
- 27 .Connector
- 28 .Constraint
- 29 .Manipulate
- 30 .Risk

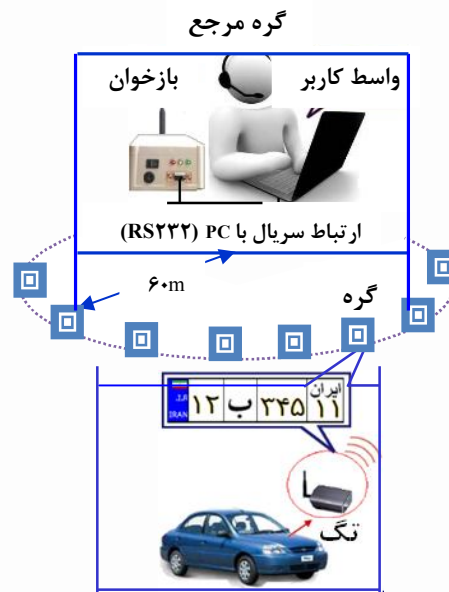
جدول ۱. ارزیابی برخی سبک‌های معماری نرم‌افزار Shaw & Garlan و دلایل عدم انتخاب آنها در توسعه واسط کاربر پیشنهادی

عنوان سبک معماری	مشخصات	مزایا	معایب	دلایل عدم انتخاب
داده محور ۱- مخزنی (غیرفعال) مؤلفه: پایگاه داده (فایل) - سرویس گیرنده رابط: پرس‌وجو - به‌هنگام‌سازی ^{۳۲} ۲- تخته‌سیاه (فعال) مؤلفه: منابع دانش ^{۳۷} - تخته‌سیاه - کنترل رابط: به‌هنگام‌سازی	- ذخیره، بازیابی و به‌هنگام‌سازی پایگاه داده توسط سرویس‌گیرندگان مستقل مخزن. - ارتباط فعال و دو طرفه بین تخته‌سیاه و منابع دانش مستقلی که با دسترسی به تخته‌سیاه به دست‌کاری داده‌ها می‌پردازند. - بهره‌گیری از تخته‌سیاه در پردازش الگو ^{۳۴} ، پردازش گفتار ^{۳۵} و کاربردهای شبکه‌ای.	- اشتراک حجم زیادی از داده‌ها. - قابلیت استفاده مجدد. - پایین بودن معیار وابستگی ^{۳۳} بین سرویس‌گیرندگان. - مقیاس‌پذیری ^{۳۳} .	- دشواری توزیع داده‌ها. - حجم بالای حافظه مورد نیاز. - احتمال ایجاد گلوگاه ^{۳۱} در روش تخته‌سیاه. - عدم پشتیبانی از منطقی پیچیده‌تر.	عدم نیاز به پایگاه داده و ذخیره‌سازی دائمی شماره پلاک‌ها در واسط کاربر پیشنهادی
جریان داده ۱- لوله و صافی مؤلفه: صافی رابط: لوله ۲- دنباله دسته‌ای مؤلفه: برنامه کامل رابط: ورودی-خروجی	- پردازش داده‌ها توسط صافی. - انتقال داده‌ها، همگام‌سازی صافی‌های فعال و بافر نمودن داده‌ها (FIFO) توسط لوله. - در روش دنباله دسته‌ای، آغاز هر انتقال پس از اتمام کامل انتقال قبلی میسر بوده و عملکرد بلادرنگ امکان‌پذیر نیست.	- سادگی ^{۴۱} ترکیب‌بندی. - قابلیت استفاده مجدد. - تغییرپذیری ^{۴۲} (سهولت جایگزینی صافی) - موازی‌سازی در روش لوله و صافی.	- ناکارآمدی در کاربردهای تعاملی ^{۳۸} . - کارایی ^{۳۹} پایین به سبب عدم تعامل مستقیم صافی‌ها با یکدیگر. - دشواری مدیریت خطا ^{۴۰} بر اثر از کار افتادن صافی‌های میانی.	مغایرت این سبک با عملکرد بلادرنگ و تعاملی واسط کاربر پیشنهادی
مؤلفه‌های مستقل ۱- پردازش‌های ارتباطی - مشتری/سرویس‌دهنده ^{۴۴} - همتا به همتا ^{۴۵} ۲- سامانه مبتنی بر رخداد - احضار ضمنی ^{۴۶} - احضار صریح ^{۴۷}	- پردازش‌های مستقلی که با پیغام‌دهی ارتباط برقرار می‌کنند و قادر به کنترل یکدیگر نیستند. - در روش مشتری/سرویس‌دهنده چندین مشتری به طور همزمان از سرویس‌دهنده درخواست سرویس دارند. - در روش همتا به همتا هر مؤلفه، دو نقش ایفا می‌کند: سرویس‌گیرنده و سرویس‌دهنده. - همه‌پختی رخداد در روش مبتنی بر رخداد	- پایین بودن معیار وابستگی مؤلفه‌ها. - قابلیت عملکرد موازی مؤلفه‌ها. - مقیاس‌پذیری. - قابلیت استفاده مجدد. - روش مشتری/سرویس‌دهنده مناسب برای کاربردهای شبکه‌ای و مبتنی بر وب. - روش مبتنی بر رخداد مناسب برای کاربردهای توزیع‌شده و Packet Switching.	- پیچیدگی. - کارایی پایین در کاربردهای تعاملی به سبب ازدیاد دفعات برقراری ارتباط و پایین بودن حجم انتقال داده در هر ارتباط. - محدود شدن کنترل اجرایی ^{۴۳} - غیر قابل پیش‌بینی بودن اثرات جانبی رخداد.	واسط کاربر پیشنهادی از نوع شبکه‌ای یا توزیع‌شده نیست.
ماشین مجازی ۱- سامانه قاعده‌مند مؤلفه: برنامه-حافظه-موتور ماشین-حالت ^{۴۹} رابط: واکنشی ^{۵۰} - ذخیره‌سازی ۲- مفسر مؤلفه: حافظه-موتور تفسیر ^{۵۱} رابط: فراخوانی روال دسترس مستقیم به حافظه ^{۵۲}	- شبیه‌سازی و تست کارکردهایی که به طور واقعی پیاده‌سازی نشده‌اند. مانند شبیه‌ساز پرواز جهت تست شرایط خطرناک. - عدم وجود ماشین واقعی جهت اجرای برنامه بر روی آن. - به‌هنگام‌سازی حالت ماشین مجازی پس از تفسیر یک دستورالعمل توسط موتور تفسیر.	- قابلیت حمل ^{۴۸} - قابلیت شبیه‌سازی سامانه‌های پیچیده.	- کارایی پایین بر اثر محاسبات اضافی در حین اجرا.	واسط کاربر پیشنهادی در تعامل مستقیم با یک سامانه RFID است که به طور واقعی پیاده‌سازی شده است.

- 31. Bottleneck
- 32. Coupling
- 33. Scalability
- 34. Pattern Processing
- 35. Speech Processing
- 36. Update
- 37. Knowledge Source (KS)
- 38. Interactive
- 39. Performance
- 40. Error Handling
- 41. Simplicity
- 42. Modifiability
- 43. Execution Control
- 44. Client-Server
- 45. Peer-to-Peer
- 46. Implicit Invocation
- 47. Explicit Invocaion
- 48. Portability
- 49. State-Machine Engine
- 50. Fetch
- 51. Interpretation Engine
- 52. Direct Memory Access (DMA)

واسط کاربر پیشنهادی

در لایه کاربرد، کاربر نهایی با استفاده از یک نرم‌افزار واسط کاربر با شبکه بی‌سیم به صورت بلادرنگ^{۵۳} ارتباط برقرار می‌نماید. از طریق ارتباط سریال بازخوان با PC (RS۲۳۲)، کاربر قادر به صدور شش فرمان بی‌سیم تا شعاع ۶۰ متری بازخوان می‌باشد. به عبارت دیگر در این شبکه، مجموعه بازخوان و PC نقش گره مرجع را ایفا می‌کند و تگ‌های نصب شده روی وسایط نقلیه متحرک نیز سایر گره‌های شبکه را تشکیل می‌دهند. در شکل ۲ نحوه کنترل بی‌سیم تردد وسایط نقلیه در این سامانه نشان داده شده است.



شکل ۲. نحوه کنترل بی‌سیم تردد وسایط نقلیه

تگ‌ها با یکدیگر در تعامل نیستند بلکه هر تگ تنها با گره مرجع در ارتباط است. در این شبکه، آدرس تگ هر وسیله نقلیه، شماره پلاک الکترونیکی آن وسیله در نظر گرفته شده است. دو نوع تعامل بین تگ‌ها و بازخوان قابل انجام است. در تعامل نوع اول، پاسخ‌دهی یک تگ به بازخوان به صورت گره به گره^{۵۴} و در تعامل نوع دوم، پاسخ‌دهی چند تگ به بازخوان به صورت چندگره^{۵۵} به گره انجام می‌گیرد. نوع دوم مربوط به فرمان پویا محیط است که با همه‌پخشی^{۵۶} از سوی بازخوان همراه است و در اجرای آن با استفاده از یک الگوریتم ضد تصادم^{۵۷} تلفیقی، پدیده برخورد تگ با تگ^{۵۸} مهار شده است.

وظایف واسط کاربر

واسط کاربر موظف است امکان صدور شش فرمان بی‌سیم را برای وسایط نقلیه با شماره پلاک ایران یا ترانزیت به این شرح فراهم آورد:

(۱) فرمان پویا محیط با هدف رویت شماره پلاک کلیه وسایط نقلیه حاضر در محیط. این فرمان راهکاری ابداعی برای شناسایی تخلفاتی چون تردد غیر مجاز در طرح زوج و فرد خودروها و تعویض غیر قانونی پلاک خودروها به شمار می‌رود.

(۲) فرمان سرکشی با هدف اطلاع از حضور خودرویی مشخص در محیط از طریق وارد کردن شماره پلاک آن.

(۳) فرمان تخصیص شناسه با هدف تخصیص بی‌سیم شماره پلاک الکترونیکی به یک خودرو.

(۴) فرمان تغییر شناسه با هدف جایگزینی بی‌سیم یک شماره پلاک الکترونیکی جدید.

(۵) فرمان حذف شناسه با هدف غیر قابل رویت نمودن خودروی مورد نظر در شبکه بی‌سیم.

(۶) فرمان درخواست شناسه با هدف اطلاع از شماره پلاک وسیله نقلیه از طریق وارد کردن شماره سریال تگ آن. جهت اطلاع از شماره پلاک الکترونیکی یک خودروی مشخص نیازی به پویا محیط نیست بلکه از طریق وارد کردن شماره سریال تگ مربوطه فقط شماره پلاک همان خودرو به رویت کاربر می‌رسد.

معماری لایه‌ای^{۵۹}

در طراحی این واسط کاربر، معماری لایه‌ای از زیرشاخه‌های سبک فراخوانی و بازگشت مورد استفاده قرار گرفته است. سبک معماری فراخوانی و بازگشت مبتنی بر تقسیم کارکرد به سلسله مراتب^{۶۰} کنترلی است که در آن یک مولفه سطح بالاتر مولفه سطح پایین‌تر را فراخوانی نموده و نتیجه‌ی اجرای عملیات برگردانده می‌شود. در معماری این واسط کاربر نیز از طریق سازماندهی سلسله مراتبی، وظایف سامانه در چهار سطح انتزاعی به نام لایه توزیع شده است. در این سبک، لایه در نقش مولفه و فراخوانی روال تعامل لایه‌ها در نقش رابط عمل می‌کند. هر لایه ضمن سرویس‌دهی به لایه بالاتر، از لایه پایین‌تر سرویس می‌گیرد [۱۴، ۱۶، ۱۷].

دلایل اجتناب از توسعه تک‌پیمانه‌ای واسط کاربر پیشنهادی عبارتند از:

(۱) افزایش اثرات جانبی تغییرات: اعمال تغییر کوچک به اقتضای نیازهای جدید مستلزم بازنویسی حجم زیادی از کد بوده و اتلاف هزینه و زمان را در پی دارد [۶، ۱۶].

(۲) پیچش^{۶۱}: پیچیدگی پیاده‌سازی و درهم‌ریختگی کدها، پیامدهایی چون دشواری خطایابی، گستردگی ارجاع، تعدد متغیرها

59. Layered Architecture

60. Hierarchy

61. Tangling

53. Real-Time

54. Point-to-Point

55. Multipoint-to-Point

56. Broadcasting

57. Anti-Collision Algorithm

58. Tag-Tag Collision

- (۲) قابلیت مخفی‌سازی اطلاعات^{۶۹} لایه‌های درونی از لایه‌های بیرونی وجود دارد.
- (۳) پیش‌بینی لایه‌ای امنیتی در قالب یک کلاس کنترلی با نام امنیت که به سبب پایین‌تر بودن معیار وابستگی معماری لایه‌ای در قیاس با معماری شیء‌گرا، افزایش قابلیت اطمینان^{۷۰} را به همراه داشته است.
- (۴) افزایش انسجام در هر مولفه و کاهش وابستگی بین مولفه‌ها از ویژگی‌های کمی ارتقاء یافته بر اثر این تلفیق محسوب می‌شوند.
- (۵) ویژگی‌های کیفی ارتقاء یافته بر اثر این تلفیق در جدول ۲ ارائه شده است.

و تعدد مسیرهای کنترلی بر اثر عدم تفکیک کد مولفه‌ها را در پی دارد [۱۴، ۶].

در گام طراحی واسط کاربر پیشنهادی از معماری لایه‌ای و در گام پیاده‌سازی آن از برنامه‌نویسی شیء‌گرا استفاده شده تا از مزایای تلفیق لایه‌بندی و شیء‌گرایی بهره‌گیری شود چراکه صرفاً بهره‌گیری از سبک معماری شیء‌گرا، معایب زیر را به همراه دارد [۱۸-۱۶] و [۲۰]:

- (۱) لزوم اطلاع هر شیء از هویت کلیه اشیایی که با آنها در تعامل است.
- (۲) تغییر مشخصات یک شیء به تغییر کلیه اشیایی می‌انجامد که به طور ضمنی آن شیء را احضار می‌کنند.
- (۳) چنانچه شیء (الف) و شیء (ب) به طور همزمان شیء (ج) را احضار کنند، تاثیر (ب) روی (ج) می‌تواند به مشکل اثر جانبی^{۶۲} غیر منتظره روی (الف) منجر شود.
- (۴) در معماری شیء‌گرا، معیار وابستگی بین ابرکلاس‌ها^{۶۴} و زیرکلاس‌ها^{۶۴} بالاست.
- همچنین محدودیت‌های برنامه‌نویسی شیء‌گرا نیز عبارتند از [۱۶، ۶]:

- (۱) دشواری مفاهیم و روش‌های مطرح در شیء‌گرایی در قیاس با برنامه‌نویسی ساخت یافته^{۶۵}.
- (۲) دشواری مجتمع‌سازی^{۶۶} کلاس‌ها در صورت ازدیاد آنها.
- به منظور غلبه بر محدودیت‌های فوق، پیاده‌سازی واسط کاربر پیشنهادی تنها با چهار کلاس انجام شده و هر کلاس با برخورداری از انسجام^{۶۷} بالا در لایه‌ای مجزا استقرار یافته است.
- محدودیت‌های سبک معماری لایه‌ای نیز عبارتند از:
- (۱) دشواری در تشخیص سطح تجرد هر لایه در حین جداسازی لایه‌ها.
- (۲) کاهش معیار کارایی معماری لایه‌ای به سبب ازدیاد لایه‌ها و سربار^{۶۸} ناشی از آنها.

به منظور مقابله با مشکلات فوق، در این مقاله تنها از چهار لایه در طراحی معماری واسط کاربر استفاده شده است. مزایای تلفیق لایه‌بندی و شیء‌گرایی در توسعه واسط کاربر پیشنهادی عبارتند از:

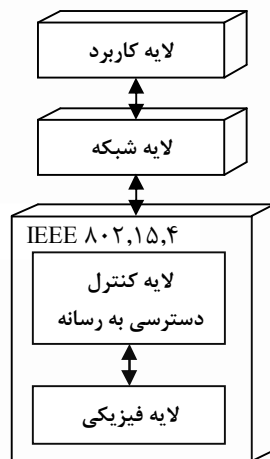
- (۱) ایجاد تغییر در هر لایه حداکثر بر یک لایه بالاتر و یک لایه پایین‌تر اثرگذار بوده و بر سایر لایه‌ها بی‌تاثیر است.

62. Side Effect
63. Super class
64. Sub class
65. Structured Programming
66. Integration
67. Cohesion
68. Overhead

69. Data Hiding
70. Reliability

جدول ۲. ویژگی‌های کیفی ارتقاء یافته بر اثر تلفیق لایه‌بندی و شی‌گرایی

نقش برنامه‌نویسی شی‌گرا		نقش معماری لایه‌ای		ویژگی‌های کیفی ارتقاء یافته در واسط کاربر پیشنهادی
توضیحات	عامل ارتقاءدهنده	توضیحات	عامل ارتقاءدهنده	
- کاهش پیچیدگی‌ها. - انعطاف‌پذیری ^{۷۱} در مقابل تغییرات. - تسهیل و تسریع تست واسط.	کپسوله‌سازی ^{۷۲}	- کاهش پیچیدگی‌های تحلیل و طراحی بر اثر لایه‌بندی. - افزایش قابلیت مدیریت ^{۷۳} کدها.	تقسیم وظایف بین ۴ لایه	بهره‌وری ^{۷۴}
- کاهش انتشار خطا به سایر نقاط نرم‌افزار واسط بر اثر مخفی‌سازی اطلاعات سایر کلاس‌ها نمی‌باشند.	مخفی‌سازی اطلاعات	- افزایش سرعت اجرا با توزیع پردازش‌ها در ۴ لایه.		
- تسهیل ارتقاء و کاهش زمان به‌روزرسانی.	کلاس‌بندی	- تسهیل ارتقاء و کاهش زمان به‌روزرسانی.	لایه‌بندی	قابلیت گسترش ^{۷۵}
- انتزاع ^{۷۶} و وراثت زمینه‌ساز افزایش قابلیت استفاده مجدد از کد.	وراثت ^{۷۷} مفهوم شیء	- عدم وابستگی لایه‌های پایین‌تر به لایه‌های بالاتر زمینه‌ساز افزایش قابلیت استفاده مجدد.	لایه‌بندی	قابلیت استفاده مجدد
- تسهیل تست واسط از طریق تست کلاس و توابع شیء - بهبود قابلیت نگهداری (تغییر و ارتقاء).	کلاس‌بندی	- افزایش بهره‌وری در خطایابی و اشکال‌زدایی کدها. - کاهش هزینه نگهداری. - تسهیل ارتقاء و کاهش زمان به‌روزرسانی. - تغییر هر لایه حداکثر به تغییر در یک لایه بالاتر و یک لایه پایین‌تر منجر شده و بر سایر لایه‌ها بی‌تاثیر است.	استقلال کارکردی لایه‌ها	قابلیت نگهداری ^{۷۸}
- پیش‌بینی کلاسی کنترل با هدف رمزنگاری اطلاعات کاربر و رمزگشایی اطلاعات بازخوان.	ایجاد کلاس امنیت	- با تشخیص مناسب سطح تجرد هر لایه معیار وابستگی بین لایه‌ها کاهش و امنیت افزایش یافته است.	وابستگی پایین لایه‌ها	امنیت ^{۷۹}



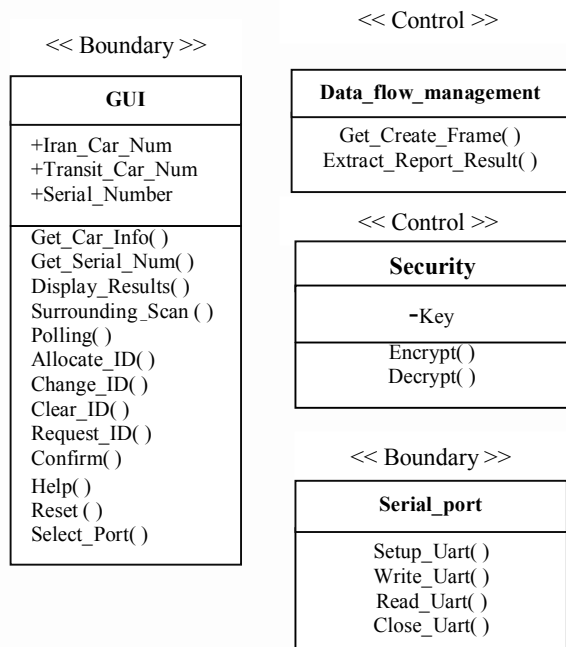
شکل ۳- الف. مدل لایه‌ای سامانه RFID فعال

در این سامانه، لایه کاربرد با استفاده از معماری لایه‌ای به پنج مولفه منطقی تقسیم‌بندی شده است. انسجام بیشینه در هر مولفه و وابستگی کمینه بین مولفه‌ها از معیارهای تجزیه^{۸۰} این سامانه به مولفه‌های مزبور به شمار می‌روند [۱۵].

در شکل ۳- الف مدل لایه‌ای سامانه RFID فعال با رویکرد بالا به پایین^{۸۱} و در شکل ۳- ب نیز لایه‌های واسط کاربر پیشنهادی نشان داده شده است.

- 71. Flexibility
- 72. Encapsulation
- 73. Manageability
- 74. Efficiency
- 75. Extensibility
- 76. Abstraction
- 77. Inheritance
- 78. Maintainability
- 79. Security
- 80. Decomposition
- 81. Top-Down Approach

انسجام بیشینه در هر کلاس و وابستگی کمینه بین کلاسها انجام گرفته است تا استقلال کارکردی هر کلاس فراهم شده و با سایر کلاسها تعامل اضافی نداشته باشد. در شکل ۴ کلاسهای واسط کاربر نشان داده شده‌اند.



شکل ۴. کلاسهای واسط کاربر

کلاسهای واسط کاربر عبارتند از:

(۱) کلاس مرزی با نام رابط کاربر گرافیکی که دریافت فرمانهای کنترلی کاربر، اعتبارسنجی اطلاعات ورودی و نمایش نتایج اجرای فرمانها را بر عهده دارد.

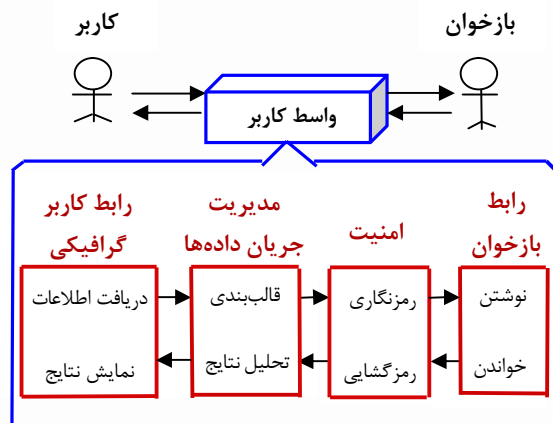
(۲) کلاس کنترلی با نام مدیریت جریان داده‌ها که ساختاردهی به اطلاعات ورودی، کنترل خطا و تحلیل نتیجه اجرای عملیات را برعهده دارد. در شکل ۵ قالببندی اطلاعات ورودی در کلاس مدیریت جریان داده‌ها نشان داده شده است.

نشانهگر اختتام (۱ بایت)	نوع خودرو (۱ بایت)	شماره پلاک (۸ بایت)	کد فرمان (۱ بایت)	نشانهگر آغاز (۱ بایت)
----------------------------	-----------------------	------------------------	----------------------	--------------------------

شکل ۵. نحوه قالببندی اطلاعات ورودی

(۳) کلاس کنترلی با نام امنیت که به منظور رمزنگاری اطلاعات ارسالی برای درگاه سریال و رمزگشایی اطلاعات دریافتی از آن پیش‌بینی شده است.

(۴) کلاس مرزی با نام درگاه سریال که تعامل با بخش ارتباط سریال UART در دستگاه بازخوان را بر عهده دارد. این کلاس



شکل ۳- ب. نمایش لایه‌های واسط کاربر پیشنهادی

اجزای لایه کاربرد به ترتیب عبارتند از:

- (۱) رابط کاربر گرافیکی^{۸۲} شامل اجزای ویژوال مانند دکمه‌ها، جعبه‌های متنی و... به منظور تعامل با کاربر.
- (۲) مدیریت جریان داده‌ها به منظور قالببندی اطلاعات ورودی و کنترل خطا.
- (۳) امنیت به منظور رمزنگاری اطلاعات دریافت شده از کاربر و رمزگشایی اطلاعات دریافت شده از بازخوان.
- (۴) رابط بازخوان به منظور تبادل اطلاعات با بازخوان.
- (۵) میان‌افزار^{۸۳} که نرم‌افزار نهفته بازخوان را دربر دارد و برقراری ارتباط بین واسط کاربر و تگ‌ها را بر عهده دارد.

در طراحی معماری واسط کاربر از چهار جزء نخست استفاده شده است چراکه وظیفه اصلی واسط کاربر، دسترسی به درگاه سریال PC و ارسال و دریافت اطلاعات برای بازخوان می‌باشد.

پایاده‌سازی شی‌گرا و معرفی کلاسها

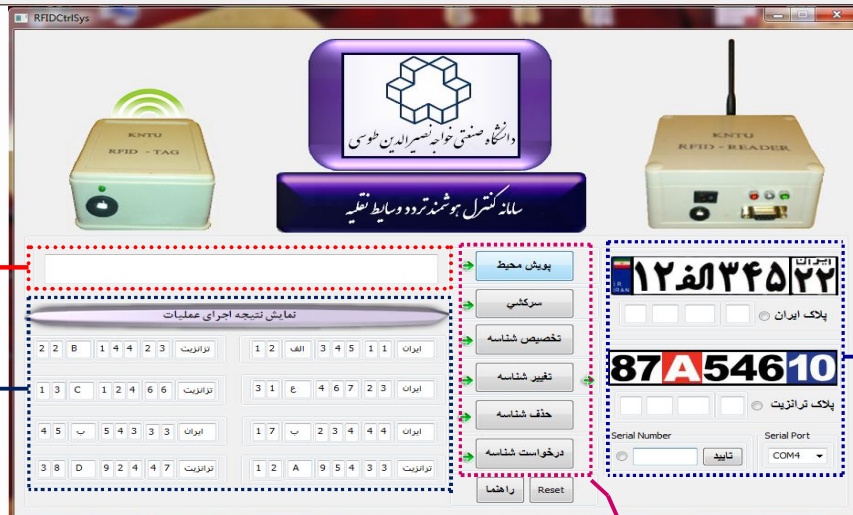
واسط کاربر، تحت ویندوز و به روش صفحه محاوره مبتنی بر کلاسهای پایه‌ی میکروسافت^{۸۴}، به زبان Visual C++ در محیط Visual Studio توسعه یافته است. از دلایل انتخاب زبان شی‌گرا، پشتیبانی از اصول کپسوله‌سازی کد و داده، انتزاع، چندریختی^{۸۵} و وراثت می‌باشد [۶]. برنامه‌نویسی شی‌گرا مبتنی بر مدل‌سازی شی‌گراست^{۸۶}. در این مقاله مدل‌سازی شی‌گرا در محیط IBM Rational Rose به انجام رسیده است. طراحی بر اساس دو اصل

82. Graphical User Interface(GUI)
83. Middleware
84. MFC Dialog Based(Microsoft Foundation Classes Dialog Based)
85. Polymorphism
86. Object Oriented Modelling (OOM)

شامل توابعی برای تنظیم پارامترهای درگاه سریال، نوشتن اطلاعات در درگاه سریال و خواندن اطلاعات از آن می‌باشد. ایجاد تغییر در سخت‌افزار خارجی متصل به درگاه سریال بر روی این کلاس بی‌تاثیر بوده و مستقل از آن طراحی شده است.

با فشردن دکمه پویش محیط، شماره پلاک ۸ خودرو در مدت ۳۱ msec به رویت کاربر می‌رسد به طوری که میانگین زمان پاسخ هر خودرو در قیاس با ۲ s زمان مورد نیاز برای خواندن

با کاهش چشمگیری همراه است [۱۹]. زمان پاسخ با وقفه تایمر تراشه تگ اندازه‌گیری شده است. در شکل ۶ امکانات واسط کاربر پیشنهادی نشان داده شده است.



نمایشگر پیغام‌های بازخوان

بخش نمایش نتایج عملیات

بخش دریافت اطلاعات ورودی

شکل ۶. امکانات واسط کاربر توسعه‌یافته

بخش صدور فرمانهای بی‌سیم

در شکل ۷ نقش واسط کاربر در عملکرد سامانه کنترل هوشمند تردد وسایط نقلیه نشان داده شده است.



شکل ۷. نقش واسط کاربر در عملکرد سامانه کنترل هوشمند تردد وسایط نقلیه

در جدول ۳ مشخصات واسط کاربر توسعه داده شده، ارائه شده است:

جدول ۳. مشخصات واسط کاربر توسعه داده شده

توضیحات	مشخصات واسط کاربر	پارامترهای ارزیابی
تعامل با کاربر و سخت افزار بازخوان متصل به درگاه سریال PC	بلادرنگ	نوع نرم افزار
—	ویندوز	بستر اجرا
تعداد لایه ها: ۴ عدد	معماری لایه ای	الگوی معماری
ترسیم مدل مورد کاربرد، مدل کلاس و مدل رفتاری	IBM Rational Rose	محیط مدل سازی
—	Visual C++	زبان برنامه نویسی
—	شی گرا	راهبرد پیاده سازی
—	MFC Dialog Based	فراروش پیاده سازی
—	Microsoft Visual Studio	محیط پیاده سازی
تست توابع شی _ تست کلاس ها	تست قطعه	روش کنترل کیفی
—	کلاس رابط کاربر گرافیکی کلاس درگاه سریال	کلاس های مرزی
—	کلاس مدیریت جریان داده ها کلاس امنیت	کلاس های کنترلی
Baud rate = ۹۶۰۰ b/sec Bit size = ۸ bits Stop bit=۱ bit No Parity	درگاه سریال PC _ استاندارد RS۲۳۲	نحوه تعامل با بازخوان
شناسایی همزمان کلیه خودروهای حاضر در میدان بازخوان میانگین زمان پویس ۸ خودرو: ۳۱ msec	عملیات پویس محیط	قابلیت ها
اطلاع از حضور خودرویی با شماره پلاک مورد نظر در محیط	عملیات سرکشی	
تخصیص بی سیم شماره پلاک الکترونیکی به یک خودرو	عملیات تخصیص شناسه	
جایگزینی بی سیم یک شماره پلاک الکترونیکی جدید	عملیات تغییر شناسه	
غیر قابل رویت نمودن خودروی مورد نظر در شبکه بی سیم	عملیات حذف شناسه	
اطلاع از شماره پلاک یک خودرو از طریق وارد کردن شماره سریال تگ آن	عملیات درخواست شناسه	کاربردها
کاربرد در پارکینگ ها، اماکن محافظت شده، جاده ها و بزرگراه ها	— شناسایی تردد غیر مجاز وسایط نقلیه — شناسایی تعویض غیرقانونی پلاک وسایط نقلیه — کنترل حمل و نقل کالا در کاربردهای تجاری — شناسایی ادوات نظامی اعم از تانک ها و ماشین های سبک و سنگین نظامی	

لایه فیزیکی

جدول ۴. مشخصات سامانه RFID فعال

مشخصات		ارزیابی سامانه RFID فعال
یکا	مقدار	
GHz	۲/۳۹۴ ~ ۲/۵۰۷	فرکانس RF
-	O-QPSK	روش مدولاسیون
kb/sec	۲۵۰	نرخ انتقال اطلاعات بی‌سیم (تعامل بازخوان و تگ)
dBm	-۹۷ ~ -۸۸	حساسیت گیرنده
dBm	-۲۸ ~ +۴/۵	توان سیگنال ارسالی
bit/sec	۹۶۰۰	نرخ انتقال اطلاعات سریال (تعامل بازخوان و PC)
mA	۲۹/۶	جریان مصرفی بازخوان
mA	۲۰/۸	جریان مصرفی تگ (در حالت فعال)
μA	۰/۸	جریان مصرفی تگ (در حالت خواب)
m	۶۰	محدوده پوشش‌دهی
mm ^۲	۵۸ × ۵۴	ابعاد برد بازخوان
mm ^۲	۵۱ × ۴۸	ابعاد برد تگ

سخت‌افزار ساخته شده بازخوان

بازخوان از طریق ارتباط سریال با PC و یک واسط کاربر، فرمانهای کاربر را دریافت نموده و آنها را به صورت بی‌سیم از طریق ریزموج^{۹۱} در باند بدون مجوز^{۹۲} ویژه‌ی کاربردهای صنعتی، علمی و پزشکی^{۹۳} به تگ‌ها تا شعاع ۶۰ متری مخابره می‌نماید و پس از دریافت پاسخ تگ‌ها، نتایج را به PC انتقال می‌دهد تا در محیط گرافیکی واسط کاربر نمایش داده شوند. در شکل ۸ اجزای سخت‌افزار بازخوان نشان داده شده است.

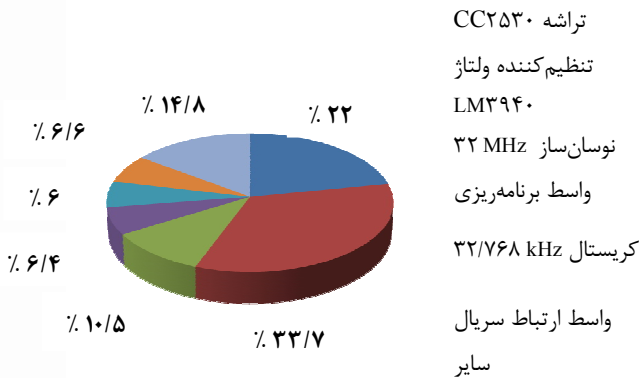
در لایه فیزیکی، انتقال بیت‌ها به صورت سیگنال الکتریکی با نرخ ارسال ۲۵۰ kb/s بر روی کانال مخابراتی با فرکانس مرکزی GHz ۲/۴۰۵ انجام می‌گیرد. استاندارد IEEE ۸۰۲،۱۵،۴ دارای ۱۶ کانال در باند فرکانسی GHz ۲/۴ با فواصل ۵ MHz می‌باشد و از مدولاسیون DSSS^{۸۷} و روش O-QPSK^{۸۸} بهره می‌گیرد. حرکت نزدیک‌شونده تگ به سمت بازخوان، فرکانس را رو به بالا شیفت می‌دهد که به اثر داپلر^{۸۹} موسوم است [۲۱، ۲۲]. استفاده از مدولاسیون طیف گسترده در این سامانه موجب شده است تا اثر داپلر در سرعت‌های بالای وسایط نقلیه نیز قابل چشم‌پوشی باشد و استقرار بازخوان در مکان‌هایی که سرعت‌گیر در آنها نصب شده است، الزامی نباشد. ساخت بازخوان و تگ فعال و برنامه‌نویسی تراشه‌ی آن دو در محیط IAR Embedded Workbench نیز توسط نویسندگان مقاله به انجام رسیده است. نوآوری طراحی و ساخت بازخوان و تگ فعال در این سامانه طراحی تک‌تراشه‌ای است. امتیاز طراحی سخت‌افزار بازخوان و تگ در این سامانه، اجتماع رادیو، ریزپردازنده و مولد اعداد تصادفی^{۹۰} در داخل یک تراشه با نام CC۲۵۳۰F۲۵۶ می‌باشد. در کارهای مشابه، یک تراشه به عنوان ریزپردازنده به تراشه دیگری به عنوان گیرنده-فرستنده رادیویی متصل شده است [۱، ۲۳]. ایجاد نویزهای فرکانس بالا بر اثر اتصال دو تراشه و افزایش ابعاد سخت‌افزار از پیامدهای نامطلوب این طراحی محسوب می‌شود. در این مقاله با بهره‌گیری از تراشه CC۲۵۳۰F۲۵۶، ضمن حذف نویزهای مزبور، اتصالات مداری و ابعاد سخت‌افزار بازخوان و تگ فعال نیز کاهش یافته است. تراشه CC۲۵۳۰F۲۵۶، یک تراشه با مصرف توان پایین به ابعاد mm ۶ × ۶ می‌باشد و برای برقراری ارتباطات بی‌سیم مطابق با استاندارد IEEE ۸۰۲،۱۵،۴ طراحی شده است. محدوده پوشش‌دهی وسیع، حساسیت گیرندگی بالا و پشتیبانی از ۱۶ کانال رادیویی از دیگر مزایای این تراشه می‌باشد. حفظ تقارن از اصول طراحی شبکه تطبیق امپدانس در بازخوان و تگ فعال محسوب می‌شود [۲۷-۲۴].

در جدول ۴ مشخصات این سامانه RFID فعال، ارائه شده است [۲۵، ۲۴]:

91. Microwave
92. Unlicensed
93. Industrial, Scientific and Medical Band (ISM)

87. Direct Sequence Spread Spectrum (DSSS)
88. Offset Quadrature Phase Shift Keying (O-QPSK)
89. Doppler effect
90. Random Number Generator(RNG)

در شکل ۱۰ نمودار توان مصرفی اجزای سخت‌افزار بازخوان نشان داده شده است.

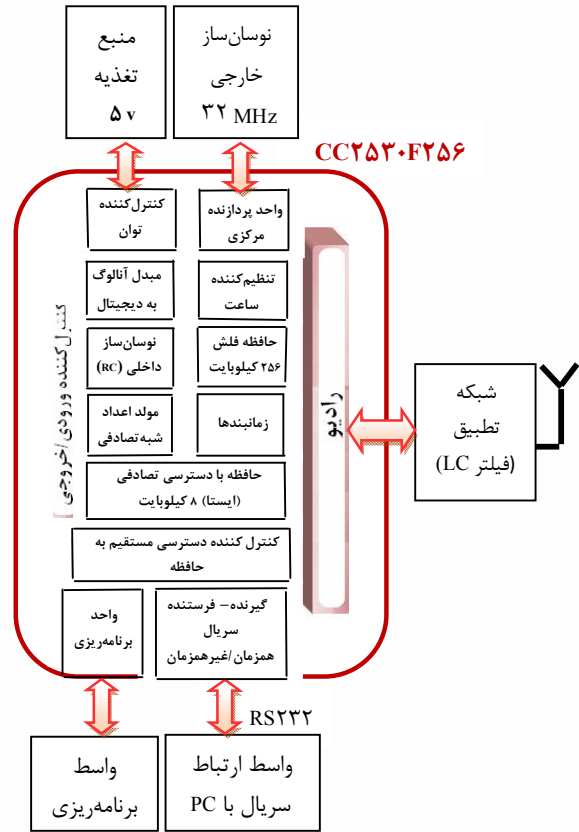


شکل ۱۰. نمودار توان مصرفی اجزای سخت‌افزار بازخوان

سخت‌افزار ساخته شده تگ فعال

محدوده پوشش‌دهی وسیع‌تر، قابلیت اجرای عملیات پیچیده‌تر و قابلیت ذخیره‌سازی اطلاعات بیشتر از مزایای تگ فعال در قیاس با تگ غیر فعال به شمار می‌رود که بهبود کارایی سامانه RFID فعال را در پی دارد. پیچیدگی مدار و مصرف بالای توان از چالش‌های طراحی سخت‌افزار تگ‌های فعال محسوب می‌شود. کلیه تگ‌های وسایط نقلیه که در این مقاله مورد استفاده قرار گرفته‌اند، سخت‌افزار یکسانی دارند و ریزپردازنده آنها با برنامه واحدی برنامه‌ریزی شده است. تگ‌های غیر فعال تنها قادرند با قرارگیری در معرض میدان الکترومغناطیسی بازخوان، شناسه خود را ارسال کنند در حالی که تگ‌های فعال ساخته شده قادر به اجرای شش فرمان بی‌سیم کارآمد و متفاوت هستند. این تگ‌ها اغلب در وضعیت خواب^{۹۵} یا ذخیره‌سازی توان^{۹۶} قرار دارند. در این وضعیت به سبب غیرفعال شدن تنظیم‌کننده ولتاژ بخش دیجیتال و نوسان‌سازهای فرکانس بالا، مصرف توان در تراشه CC2530F256 کاهش می‌یابد. قرارگیری در وضعیت خواب، نوعی سازوکار ذخیره توان تگ‌ها در لایه فیزیکی به شمار می‌رود. جریان مصرفی در وضعیت خواب برابر با $0.8 \mu A$ می‌باشد. هر تگ به محض دریافت بی‌سیم بسته اطلاعات، به وضعیت فعال منتقل می‌شود. جریان مصرفی در وضعیت فعال $20.8 mA$ می‌باشد [۲۴، ۲۵].

عدم نیاز به منبع تغذیه مجزا از امتیازات طراحی مدار تغذیه این تگ فعال است. ولتاژ تغذیه مورد نیاز، مستقیماً از باتری ۱۲ ولتی وسیله نقلیه‌ای که تگ در آن نصب شده است، تامین می‌شود. در شکل ۱۱ اجزای سخت‌افزار تگ فعال نشان داده شده است.



شکل ۸. اجزای سخت‌افزار بازخوان

استاندارد RS232 برای شبکه‌های ساده با یک بازخوان مناسب می‌باشد. در صورت استفاده از چندین بازخوان، استاندارد RS485 با خط ارسال تفاضلی، انتخاب مناسب‌تری محسوب می‌شود [۲۸]. با استفاده از نرم‌افزار SmartRF Studio V7، نرخ خطای^{۹۴} بسته‌ها و شاخص قدرت سیگنال دریافتی در فواصل گوناگون اندازه‌گیری شده است [۲۶، ۲۷].

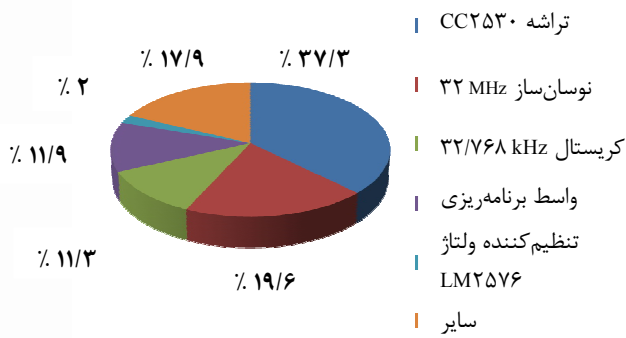
در شکل ۹ بازخوان پس از ساخت، نشان داده شده است.



شکل ۹. بازخوان پس از ساخت

95. Sleep Mode
96. Power-Saving

94. Packet Error Rate (PER)



شکل ۱۳. نمودار توان مصرفی اجزای سخت‌افزار تگ فعال

نتیجه‌گیری

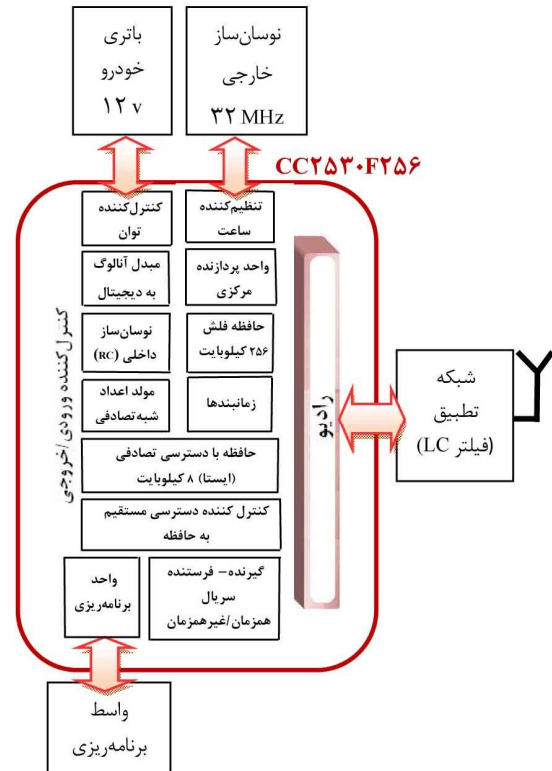
با توجه به گسترش روزافزون کاربردهای RFID فعال، در این مقاله، دو بخش مهم در طراحی و پیاده‌سازی یک سامانه RFID فعال در فرکانس 2/4 GHz مورد بررسی قرار گرفت. در بخش نرم‌افزار، توسعه چندلایه و شی‌گرایی واسط کاربر تشریح شد که قابلیت صدور شش فرمان بی‌سیم را در حوزه تردد وسایط نقلیه متحرک داراست. ویژگی‌های کمی ارتقاء یافته در توسعه این واسط عبارتند از: کاهش زمان پاسخ، افزایش انسجام در هر مولفه، کاهش وابستگی بین مولفه‌ها و کاهش حجم حافظه مورد نیاز. ویژگی‌های کیفی ارتقاء یافته نیز عبارتند از: کارایی، قابلیت گسترش، قابلیت نگهداری، قابلیت استفاده مجدد و امنیت. در بخش سخت‌افزار، طراحی و ساخت سخت‌افزار بازخوان و تگ فعال بررسی شد. با طراحی تک‌تراشه‌ای و اجتماع رادیو، مولد اعداد تصادفی و ریزپردازنده داخل یک تراشه، کاهش اتصالات مداری، حذف نویزهای فرکانس بالا و کاهش توان مصرفی سخت‌افزارهای بازخوان و تگ فعال محقق شده است.

سپاسگزاری

با سپاس فراوان از جناب آقای مهندس فرزاد نوریان و سرکار خانم مهندس سمانه وحیدی.

مرجع‌ها

- [1] X. Huang, G. Zhang, Q. Sun, and D. Qing, "Container Transportation System using 2.45 GHz Active RFID technology," International Conf. on Remote Sensing, Environment and Transportation Engineering. Nanjing, pp. 3030-3033, June 2011.
- [2] P. Anuradha, and R. Sendhilkumar, "Design and Implementation of ZigBee-RFID Based Vehicle Tracking," International Conf. on Sustainable Energy and Intelligent Systems. Chennai, pp. 689-694, July 2011.



شکل ۱۱. اجزای سخت‌افزار تگ فعال

در شکل ۱۲ تگ فعال پس از ساخت، نشان داده شده است.



شکل ۱۲. تگ فعال پس از ساخت

در شکل ۱۳ نمودار توان مصرفی اجزای سخت‌افزار تگ فعال نشان داده شده است.

- [16] A. Kumar, "Software Architecture Styles a Survey," *International Journal of Computer Applications*, Vol. 87, No. 9, February 2014.
- [17] M. Galster, A. Eberlein and M. Moussavi, "Systematic selection of software architecture styles," *Software IET*, Vol. 4, pp. 349-360, October 2010.
- [18] E. Majidi, M. Alemi and H. Rashidi, "Software Architecture: A Survey and Classification," *International Conf. on Communication Software and Networks*. Singapore, pp. 454-460, February 2010.
- [19] C.Y. Chen and P.W. Tsai, "A design of multi-heterogeneous system using SOA and RFID middleware platform," *International Conf. on Machine Learning and Cybernetics*. Qingdao, pp. 2956-2961, July 2010.
- [20] S. A. Julia, N. Snehalatha and P. Rodrigues, "Comparison of Software Architecture Styles Using Quality Attributes," *International Journal of Science and Modern Engineering software Engineering*, Vol. 1, No. 4, pp. 638-653, March 2013.
- [21] Zh. Xiaoqiang and T. Manos, "Applications of Fast-Moving RFID Tags in High-speed Railway Systems," *International Journal of Engineering Business Management*, Vol. 3, pp. 27-31, 2011.
- [22] N. Alam, A. T. Balaie and A.G. Dempster, "Dynamic Path Loss Exponent and Distance Estimation in a Vehicular Network Using Doppler Effect and Received Signal Strength," *International Conf. on Vehicular Technology*. Ottawa, pp. 1-5, September 2010.
- [23] W.J. Yoon, S.H. Chung, and S.J. Lee, "Implementation and performance evaluation of an active RFID system for fast tag collection," *Journal of computer communications*, Vol. 31, No. 17, pp. 4107-5116, November 2008.
- [24] Texas Instruments, "CC253X System-on-Chip Solution for 2.4 GHz IEEE 802. 15. 4 and ZigBee Applications (User's Guide)," 2009. <http://www.ti.com>
- [25] Texas Instruments, "A True System-on-Chip Solution for 2.4 GHz IEEE 802. 15. 4 and Zigbee Applications," 2009. <http://www.ti.com>
- [26] D. Moeinfar, H. Shamsi and F. Nafar, "Design and Implementation of a low-Power Active RFID for Container Tracking at 2.4 GHz Frequency," *Computer Science & Communications*, Vol. 2, No. 2, pp. 13-22, April 2012.
- [27] D. Moeinfar and H. Shamsi, "A Low-Collision CSMA-Based Active RFID for Tracking Applications," *Wireless Personal Communications*, Vol. 71, No. 4, pp. 2827-2847, August 2013.
- [28] V. Dashevsky and B. Sokolov, "New concept of RFID reader networks structure: hardware and software architecture," *International Conf. on Ultra Modern Telecommunications & Workshops*. St. Petersburg, pp. 1-4, October 2009.
- [3] Z. Min, L. Wenfeng, L. Bin and R. Xia, "A RFID-based Material Tracking Information System," *International Conf. on Automation and Logistics*. Jinan, pp. 2922-2926, August 2007.
- [4] W. Su, N. Alchazidis, and T.T. Ha, "Multiple RFID Tags Access Algorithm" *IEEE Trans. Mobile Computing*, Vol. 9, No. 2, pp. 174-187, 2010.
- [5] S.H. Gwon, S.C. Oh, N. Huang, and S.K. Hong, "Advanced RFID Application for a Mixed-Product Assembly Line" *International Journal of Advanced Manufacturing Technology*, Vol. 56, No. 1-4, pp. 377-386, September 2011.
- [6] R. Pressman, *Software Engineering: A Practitioner's Approach*. McGraw-Hill, pp. 328-331, January 2011.
- [7] A. Feizi, and Y.W. Chui, "Usability of User Interface Styles for Learning a Graphical Software Application," *International Conf. on computer & information science (ICCIS)*. Kuala Lumpur, pp. 1089-1094, June 2012.
- [8] Q. Xie, "Developing cost-effective model-based techniques for GUI testing," *International Conf. on software engineering (ICSE'06)*, Shanghai, pp. 997-1000, August 2006.
- [9] N.J. Nunes, and J.F.e. Cunha, "Wisdom – A UML based architecture for interactive systems" *Design, Specification and Verification of Interactive Systems*, Vol. 1946, pp. 191-205, May 2001.
- [10] F. Montero, M.D. Lozano and P. Gonzalez, "IdealXML: an Experience-Based Environment for User Interface Design and pattern manipulation" *Technical report DIAB-05-01-4*, University of Castilla-La Mancha, 2005.
- [11] F. Montero, V. Lopez-Jaquero, J. Vanderdonck, P. Gonzalez, M. Lozano and Q. Limbourg, "Solving the Mapping Problem in User Interface Design by Seamless Integration in IdealXML," *Design, Specification and Verification of Interactive Systems*, Vol. 3941, pp. 161-172, 2006.
- [12] L.V. Massawe, J. Kinyua, and F. Aghdasi, "An Implementation of a Multi-Agent Based RFID Middleware for Asset Management System Using the Jade Platform," *International Conf. on IST-Africa*. Durban, pp. 1-8, May 2010.
- [13] I. Dzafic, J. Sofo, E. Halilovic, N. Lecek and M. Music, "Object-oriented database and user interface design," *International Conf. on EuroCon*, IEEE. Zagreb, pp. 558-563, July 2013.
- [14] L. Bass, P. Clements and R. Kaszman, *Software Architecture in Practice (3rd edition) (SEI Series in Software Engineering)*. Addison-Wesley Professional, pp. 328-331, October 2012.
- [15] L. Dobrica and E. Niemela, "A Survey on Software Architecture Analysis Methods," *IEEE Trans. Software Engineering*, Vol. 28, No. 7, pp. 638-653, July 2002.

