



Original Research Paper

Implementation of LDUT-Based Redundancy and Frame Structure Management in CAN Bus Space Applications

Morteza Farhid^{1*} , Samira Mirmazhari Anvar², Moharram. Shameli³ , and Hossain Beheshti Beyrami⁴

1-3. Space Trusters Research Institute, Iranian Space Research Center, Tabriz, Iran

4. Space Trusters Research Institute, Iranian Space Research Center, Tabriz, Iran

ARTICLE INFO

Article History:

Received 29 January 2023

Revised 11 May 2023

Accepted 21 May 2023

Available Online 21 May 2023

Keywords:

Thermal barrier coating

Gas turbine

Roughness

Numerical simulations

MPLAB software

ABSTRACT

This paper discusses how to implement the CAN bus protocol based on the ECSS standard in the space applications of satellites and various missions. The CAN protocol frame consists of the frameworks, arbitration and control, data, cyclic redundancy, and verification sections. The arbitration and control section are responsible for controlling the data flow. There are two standards for these two parts, which satellites usually use the developed CAN standard according to which the LDUT protocol is designed, due to the need to use the LDUT protocol for big data transmission. Also, to manage the redundancy of the CAN bus, the implementation conditions of the parameters of each node will be checked. Also, this protocol is implemented on the reaction wheel condition with microcontroller hardware and related converter considerations in MPLAB software environment and is approved by Microchip Analyzer CAN separately and OBC simulator as a quadruple group with the correct function of the approved bus redundancy.

* Corresponding Author's E-mail: m.farhid@isrc.ac.ir

How to Cite this Article:

M. Farhid, S. Mirmazhari Anvar, M. Shameli, and H. Beheshti Beyrami, "Implementation of LDUT-Based Redundancy and Frame Structure Management in CAN Bus Space Applications," *Journal of Technology in Aerospace Engineering*, Vol. 8, No. 1, pp. 67-78, 2024, (in Persian), <https://doi.org/10.22034/jtae.2024.8.1.5>.

COPYRIGHTS

Authors retain the copyright and full publishing rights.

Published by Aerospace Research Institute. This article is an open access article licensed under the [Creative Commons Attribution 4.0 International \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/).





پیاده‌سازی و ارزیابی مدیریت افزونگی در گذرگاه CAN بر پایه قرارداد LDUT

مرتضی فرهید^{۱*}، سمیرا میر مظهری انور^۲، محرم شاملی^۳، و حسین بهشتی بیرامی^۴

۱-۳- استادیار، پژوهشکده رانشگرهای فضایی، پژوهشگاه فضایی ایران، تبریز، ایران
۴- دانشجوی دکتری، پژوهشکده رانشگرهای فضایی، پژوهشگاه فضایی ایران، تبریز، ایران

چکیده

در این مقاله جزئیات و نتایج اجرای استاندارد پروتکل ارتباطی گذرگاه CAN را برای ماموریت‌های فضایی ارائه می‌کنیم. فریم پروتکل ارتباطی CAN از بخش‌های شروع فریم، داوری و کنترل، داده، کد افزونگی چرخه‌ای و تأیید تشکیل شده‌است. بخش داوری و کنترل، وظیفه کنترل جریان داده را بر عهده دارد. برای این دو بخش دو استاندارد وجود دارد که در ماهواره‌ها معمولاً با توجه به نیاز به استفاده از پروتکل LDUT برای انتقال داده‌های بزرگ، از استاندارد توسعه‌یافته CAN استفاده می‌شود و پروتکل LDUT مطابق آن طراحی شده‌است. همچنین برای مدیریت افزونگی گذرگاه CAN، شرایط پیاده‌سازی پارامترهای هر گره بررسی شده و در نهایت این پروتکل بر روی عملگر کنترل وضعیت چرخ عکس‌العملی با ملاحظات سخت‌افزاری میکروکنترلر و مبدل مربوطه در محیط نرم‌افزاری MPLAB پیاده‌سازی شده‌است. و توسط آنالایزر CAN شرکت میکروچیپ به صورت مجزا و شبیه ساز OBC به صورت گروه چهارتایی با کارکرد صحیح افزونگی گذرگاه در کنار دیگر گره‌های ماهواره موردتصدیق قرار می‌گیرد.

اطلاعات مقاله

ناریخچه مقاله:

دریافت ۹ بهمن ۱۴۰۱
بازنگری ۲۱ اردیبهشت ۱۴۰۲
پذیرش ۳۱ اردیبهشت ۱۴۰۲
اولین انتشار ۳۱ اردیبهشت ۱۴۰۲

واژه‌های کلیدی:

پروتکل CAN
استاندارد ECSS
پروتکل LDUT
آنالایزر CAN Bus
نرم‌افزار MPLAB

* پست الکترونیکی نویسنده مسئول: m.farhid@isrc.ac.ir

How to Cite this Article:

M. Farhid, S. Mirmazhari Anvar, M. Shameli, and H. Beheshti Beyrami, "Implementation of LDUT-Based Redundancy and Frame Structure Management in CAN Bus Space Applications," *Journal of Technology in Aerospace Engineering*, Vol. 8, No. 1, pp. 67-78, 2024, (in Persian), <https://doi.org/10.22034/jtae.2024.8.1.5>.

COPYRIGHTS

Authors retain the copyright and full publishing rights.

Published by Aerospace Research Institute. This article is an open access article licensed under the [Creative Commons Attribution 4.0 International \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/).



کافی را با هزینه کم فراهم کند. چندین دهه پیش، نیازهای مشابه، صنعت خودروسازی را در اواخر دهه ۱۹۸۰ به سمت توسعه و پذیرش گذرگاه شبکه (CAN) که در حال حاضر همه جا در وسایل نقلیه و اتوماسیون صنعتی استفاده می‌شود، سوق داده بود.

شرکت اس‌اس‌تی‌ال^۸ برای اولین بار در سال ۱۹۹۸ با FASat-Bravo استفاده از CAN را در فضایما معرفی کرد و به دنبال آن UoSAT-12 در سال ۱۹۹۹ استفاده شد [۶]. شرکت اس‌اس‌تی‌ال یک اینترفیس الکتریکی استاندارد داخلی بخاطر هر ماژول با یک گذرگاه CAN دو طرفه برای مأموریت SNAP-1 که در سال ۲۰۰۰ راه اندازی شده بود، ایجاد کرده که در آن عملیات تله‌متری و فرمان از راه دور، انتقال داده و درجه خودمختاری زیر سیستم^۹ انجام می‌گردید.

CAN در حال حاضر معمولاً در ماهواره‌های تولید شده توسط اس‌اس‌تی‌ال استفاده می‌شود. همچنین مأموریت ماه SMART-1 که در سال ۲۰۰۳ پرتاب شد، اولین ماهواره ایسا بود که گذرگاه CAN را تجمیع کرد [۷]. از دو باس، یکی به عنوان گذرگاه سیستم و دیگری به عنوان گذرگاه محموله استفاده گردید. CAN به عنوان یک گذرگاه فرمان و کنترل مناسب توسط بسیاری دیگر نیز شناسایی شده است [۸-۱۰].

همانطور که بیان گردید، مأموریت‌های قبلی این را به اثبات رسانده‌اند که گذرگاه CAN را می‌توان با موفقیت در فضا مورد استفاده قرار داد. با این حال، این پیاده‌سازی‌ها و تراکنش داده‌ها با برنامه‌های نرم‌افزاری داخلی، پیاده‌سازی‌های سفارشی بوده‌اند و فقط بر تبادل داده‌های سطح پایین تمرکز دارند. از نظر مدل اتصال سیستم‌های باز^{۱۰}، استاندارد CAN فقط لایه‌های فیزیکی و پیوند داده را مشخص می‌کند. نیاز به یک پروتکل سطح بالاتر عمومی در بالای گذرگاه CAN در اوایل صنعت نیز تشخیص داده شد. بنابراین تعدادی پشته پروتکل^{۱۱} (مجموعه‌ای از پروتکل‌ها با امکان شبکه شدن برای مقصد خاص) ایجاد شد که CANopen رایج‌ترین آنها برای اتوماسیون صنعتی بود.

ایسا درگیر اتخاذ چنین پروتکل سطح بالاتری برای کاربردهای فضایی شده‌است که منجر به استاندارد شدن شد که تحت همکاری اروپایی برای استانداردسازی فضایی^{۱۲} در می ۲۰۱۵ منتشر

۱ مقدمه

گذرگاه‌های استاندارد مختلفی برای تبادل داده بین زیرسیستم‌ها در کاربردهای نظامی و فضایی توسعه یافته است. یکی از این گذرگاه‌ها، گذرگاه استاندارد انتقال داده MIL-STD-1553 است. این گذرگاه با قابلیت اطمینان نسبتاً بالا برای جابجایی داده‌ها بین زیرسیستم‌های ماهواره مورد استفاده قرار می‌گیرد و زمان اختراع آن به دهه ۱۹۷۰ (با کاربرد در هواپیماهای نظامی) برمی‌گردد [۱]. از سوی دیگر، گذرگاه MIL برای پیاده‌سازی نسبتاً پیچیده بوده و بارها بکارگیری آن به شدت محدود شده‌است [۲]. در ادامه‌ی این روند، یک شبکه مدرن‌تر در کلاس فضایی،^۱ SpaceWire، توسعه یافته و به طور رسمی در سال ۲۰۰۳ تحت نظارت ایسا^۲، استاندارد شد با قابلیت ارسال/دریافت داده‌های بسیار سریع‌تر در کنار کاهش پیچیدگی سخت‌افزاری، اجازه انعطاف‌پذیری بیشتری را در طراحی شبکه فراهم نموده و در صنعت فضایی و در مأموریت‌های مختلفی از ایسا، ناسا^۳ و جاکسا^۴ استفاده می‌گردید [۳].

برای ماهواره‌های کوچک کم هزینه، به‌ویژه برای ماهواره‌های مکعبی^۵، انتخاب معمول برای گذرگاه سیستم، گذرگاه I2C^۶ بوده است. این پروتکل در سال ۱۹۸۲ برای اتصال مدار مجتمع‌های ۷ جانی با سرعت پایین به پردازنده‌ها و میکروکنترلرها طراحی شده و معمولاً کاربرد آن جایی مطرح است که سادگی و هزینه ساخت پایین از سرعت مهمتر باشد. در واقع استفاده از I2C قبلاً در برنامه‌های ماهواره‌های مکعبی مدنظر بوده و در اکثر مأموریت‌های ماهواره‌های مکعبی نیز استفاده شده بود [۴]. اگرچه گذرگاه I2C به عنوان گذرگاه استاندارد قابل قبول برای ماهواره‌های مکعبی مطرح بود، اما بعدها به عنوان یک منبع خرابی شناخته شده در شکست مأموریت‌ها نیز گزارش گردید [۵]. یک عیب عمده این پروتکل این است که خرابی یا مشکل در نرم‌افزار یک گره می‌تواند باعث توقف کل گذرگاه شده و ارتباط بین سیستمی را غیرممکن کند. به عنوان یک اقدام جبرانی، جداکننده‌های گذرگاه یا ریست‌های دوره‌ای نیز در چند طرح ماهواره‌های مکعبی استفاده گردیده‌اند.

با فشار کاهش هزینه‌های پروژه‌های فضایی، صنعت فضایی به دنبال سیستم‌های گذرگاهی جابگزین بود که قابلیت اطمینان

6. Inter-Integrated Circuit

7. Integrated Circuit

8. Surrey Satellite Technology Ltd (SSTL)

9. Sub-System Autonomy

10. Open Systems Interconnection (OSI)

11. Protocol Stack

12. European Cooperation for Space Standardization (ECSS)

۱. یک رابط سریال نقطه به نقطه خود مدیریتی است که با ارائه یک پروتکل کاربر SpaceWire منطقی، سرعت بالا و توان مصرفی کم را ارائه می‌دهد.

۲. European Space Agency (ESA)

۳. National Aeronautics and Space Administration (NASA)

۴. Japan Aerospace Exploration Agency (JAXA)

۵. CubeSat

نمی‌شود. پشته‌های پروتکل‌های مختلفی برای رسیدگی به این مسائل ایجاد شده‌اند. الزامات CANopen که توسط سازمان CiA^۲ منتشر شده، یکی از گسترده‌ترین پروتکل‌ها برای کاربردهای صنعتی است.

CANopen یک پروتکل ارتباطی و مشخصات پروفایلی دستگاه است. این طرح شامل یک آدرس‌دهی، چندین پروتکل ارتباطی کوچک و یک لایه کاربردی است که توسط پروفایل دستگاه تعریف شده‌است. پروتکل‌های ارتباطی از مدیریت شبکه، نظارت بر دستگاه و ارتباط بین گره‌ها از جمله یک لایه انتقال ساده برای بخش‌بندی/تقسیم‌بندی پیام پشتیبانی می‌کنند. اهم مواردی که در ویژگی‌های CAN استاندارد وجود ندارد و در استاندارد جدید ECSS شرح داده شده شامل موارد زیر می‌شود [۱۲]:

- مشخصات لایه فیزیکی گذرگاه CAN برای برنامه‌های کاربردی ماهواره‌ها،
- یک پروتکل لایه بالاتر عمومی (CANopen) برای استفاده از گذرگاه CAN در برنامه‌های ماهواره‌ها
- تکنیک‌های انتقال همزمان داده‌ها از طریق گذرگاه CAN برای برنامه‌های کنترل زمان واقعی بر اساس پروتکل CANopen (اختیاری)
- توزیع زمان بر روی گذرگاه CAN برای کاربردهای ماهواره‌ها بر اساس پروتکل CANopen
- تخصیص شناسه فریم CAN bus
- مدیریت افزونگی گذرگاه CAN

استاندارد فضایی ECSS-E-ST-50-15C بیانگر پروتکل توسعه گذرگاه CAN می‌باشد. این استاندارد مشخصات اصلی شبکه CAN را تغییر نمی‌دهد و با ISO 11898-1/-2:2003 مطابقت دارد. این استاندارد الحاقات پروتکل مورد نیاز برای برآوردن الزامات خاص ماهواره را تعریف می‌کند؛ بنابراین، فرستنده‌های CAN مورد استفاده در کاربردهای فضایی باید مشخصات الکتریکی کلیدی استاندارد ISO 11898-2 را داشته باشند. علاوه بر این، آن‌ها باید بتوانند در محیط پرتوهای کیهانی موجود در فضا کار کرده، از افزونگی سیستم پشتیبانی نموده و همچنین الزامات گذرگاه CAN فضایی برای تشخیص عیب (خطا) را نیز برآورده می‌کند.

شد [۱۱]. این استاندارد در کنار اجرای ماموریت EXOMARS که در مارس ۲۰۱۶ راه اندازی شد، توسعه یافت. EXOMARS اولین و در عین حال تنها ماموریتی است که این پروتکل را در فضا به کار برده است. توضیح این که متاسفانه با وجود پیشنهاد ساختار CAN طبق استاندارد ECSS، اکثر برنامه‌های فضایی در کشور صرفاً به CAN استاندارد با تغییرات اندک بسنده کرده و ساختار فریم مبتنی بر پروتکل انتقال واحدهای داده بزرگ CAN LDUT و به قابلیت‌های آن خصوصاً در حالت تجمیع زیر سیستم‌ها توجهی نمی‌شود.

در این مقاله ما نکات استفاده در اجرای پروتکل ECSS (LDUT) CAN برای ماهواره‌های بومی با هدف افزایش قابلیت اطمینان و مدولار بودن ماموریت‌های آینده ماهواره‌ها را با جزئیات ارائه می‌دهیم. ابتدا در بخش دوم، ویژگی‌های گذرگاه CAN، پشته CAN open و نکات ضعف و قوت آنها را با اولویت پروتکل LDUT بررسی می‌کنیم. سپس در بخش سوم مدیریت افزونگی گذرگاه CAN مبتنی بر استاندارد مذکور ذکر شده و فرآیند شروع و انتخاب گذرگاه توضیح داده خواهد شد. در ادامه و در بخش چهارم ساختار فریم CAN مبتنی بر LDUT که معمولاً مناسب با طول واحدهای مختلف که از ۸ بایت بیشتر است، بررسی خواهد گردید. بخش پنجم شامل الگوریتم انتقال داده با استفاده از پروتکل مذکور بوده و در نهایت در بخش ششم موارد بحث شده در گره چرخ عکس‌عملی که عملگر زیرسیستم کنترل وضعیت ماهواره است با در نظر گرفتن ملاحظات سخت‌افزاری و نرم‌افزاری پیاده‌سازی شده و توسط ماژول آنالایزر CAN و شبیه‌ساز OBC صحت‌سنجی خواهد شد.

۲ مزایای استفاده از CAN در برنامه‌های

فضایی

بطور کلی، چندین گره را می‌توان به یک CAN اضافه کرد. این عمل به طور قابل توجهی هزینه‌های سیستم/کابل کشی را کاهش می‌دهد و در عین حال راهی مقرون به صرفه برای بهبود عملکرد را ارائه می‌دهد. طبق دستورالعمل اتحادیه فضایی اروپا، وزن و جرم ماهواره را می‌توان با این نوع سیم کشی ۱۰ تا ۱۸ درصد کاهش داد. استاندارد CAN شامل کارهای لایه بالاتر پروتکل، مانند کنترل جریان، آدرس‌دهی دستگاه و انتقال بلوک‌های بزرگ داده (دیتا)

۳ مدیریت افزونگی گذرگاه CAN

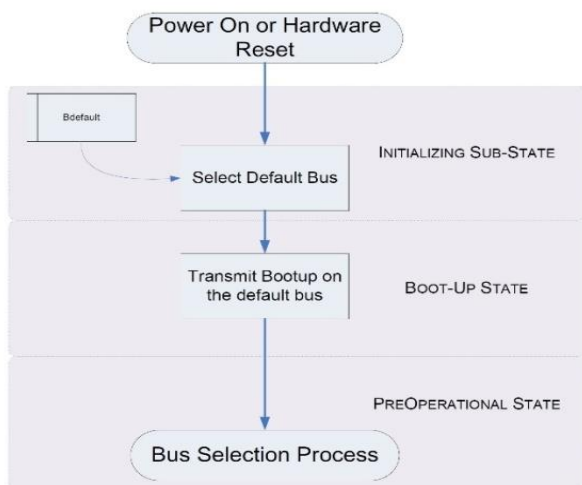
برای مدیریت افزونگی سرد گذرگاه CAN، هر گره باید پارامترهای زیر را پیاده‌سازی کند:

- زمان ضربان قلب مصرف کننده^۱: این پارامتر، مشخص کننده بیشترین زمانی است که یک گره برای دریافت ضربان قلب مدیر گذرگاه باید منتظر بماند.
- زمان ضربان قلب تولید کننده: این پارامتر، مشخص کننده بیشترین زمان میان دو ضربان قلب متوالی تولید شونده توسط مدیر گذرگاه است تنها مدیر گذرگاه ملزم به پیاده‌سازی آن است.
- گذرگاه پیش فرض^۲: به گذرگاه پیش فرضی که گره‌ها و یا مدیر گذرگاه پس از روشن شدن و یا راه‌اندازی مجدد، توسط هر گره و یا مدیر گذرگاه به عنوان گذرگاه فعال انتخاب می‌شود، گذرگاه پیش فرض گفته می‌شود.
- ضامن زمان^۳: این پارامتر، مشخص کننده تعداد دفعاتی است که یک گره قبل از تغییر گذرگاه، منتظر ضربان قلب مدیر گذرگاه می‌ماند.
- ضامن گره^۴: این پارامتر، مشخص کننده تعداد دفعاتی است که یک گره می‌تواند در صورت عدم دریافت ضربان قلب مدیر گذرگاه، گذرگاه خود را تغییر دهد.
- ضامن شمارنده^۵: شمارنده تعداد دفعاتی که هر گره گذرگاه خود را تغییر می‌دهد، ضامن شمارنده نام دارد. پس از هر بار تغییر گذرگاه، به مقدار آن یک واحد افزوده می‌شود.

۳.۱ پروسه شروع

شکل ۱ روند پروسه شروع را برای گره‌های شبکه CAN نشان می‌دهد. روند شروع به شرح زیر می‌باشد:

پس از روشن شدن و یا راه‌اندازی مجدد، هر گره باید باتوجه به پارامتر گذرگاه پیش فرض^۶، یکی از گذرگاه‌ها را به عنوان گذرگاه فعال انتخاب کند. پس از انتخاب گذرگاه فعال، هر گره باید پیامی با عنوان پیام راه‌اندازی را بر روی گذرگاه فعال منتشر کند. این پیام تنها برای مدیر گذرگاه قابل دریافت است و دیگر گره‌ها هنگام دریافت این پیام، آنرا نادیده می‌گیرند. پس از ارسال پیام راه‌اندازی، هر گره وارد حالت پیش‌عملکردی شده و پروسه انتخاب گذرگاه شروع می‌شود.



شکل ۱- روند شروع شبکه [۹]

۳.۲ پروسه انتخاب گذرگاه

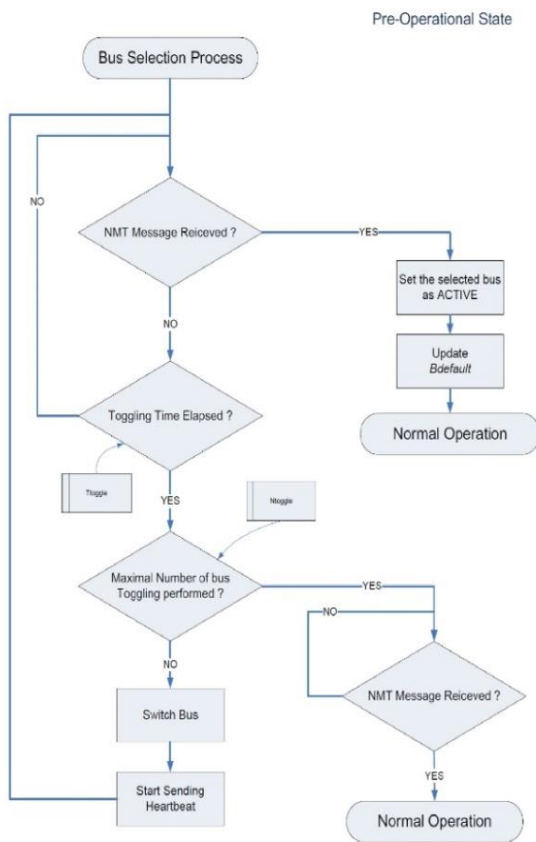
شکل ۲ روند پروسه انتخاب گذرگاه را برای گره‌های شبکه CAN نشان می‌دهد. در این پروسه، گره‌ها در وضعیت پیش‌عملکردی قرار دارند. روند انتخاب گذرگاه به شرح زیر می‌باشد:

در ابتدا هر گره منتظر دریافت ضربان قلب مدیر گذرگاه می‌ماند. در هنگام دریافت ضربان قلب مدیر گذرگاه، گره دریافت کننده، گذرگاه را به عنوان گذرگاه فعال انتخاب کرده و مقدار گذرگاه پیش‌فرض را بروزسانی کرده و سپس در حالت عملکرد عادی قرار می‌گیرد. در صورتی که در مدت زمان مشخص (زمان ضربان قلب مصرف کننده) گره قادر به دریافت ضربان قلب مدیر گذرگاه نباشد و همچنین تعداد دفعاتی که گره می‌تواند منتظر ضربان قلب مدیر گذرگاه بماند (Toggle) به پایان رسیده باشد، گره وارد مرحله تغییر مجدد گذرگاه می‌شود. در صورتی که Toggle به بیشترین حد مجاز خود نرسیده باشد، گره در حالت انتظار برای دریافت ضربان قلب می‌ماند. در صورتی که گره در مرحله تغییر مجدد گذرگاه قرار گیرد و تعداد مجاز دفعات تغییر گذرگاه (Ntoggle) تمام نشده باشد، گذرگاه را تغییر داده و روند انتخاب گذرگاه از سر گرفته می‌شود. پس از تغییر گذرگاه، گره ضربان قلب خود را بر روی گذرگاه ارسال می‌کند تا مدیر گذرگاه متوجه این تغییر شود. توجه شود که در هر بار تغییر گذرگاه، مقدار Toggle به مقدار اولیه خود باز می‌گردد. درحالی‌که مقدار Ntoggle در کل مأموریت ثابت خواهد بود و پس از عبور از حد مجاز تنها در صورتی مقدار آن به حالت اولیه باز می‌گردد که از جانب مدیر گذرگاه دستور تغییر Ntoggle دریافت شود. در صورتی که تعداد دفعات تغییر گذرگاه از حد مجاز گذشته باشد، گره

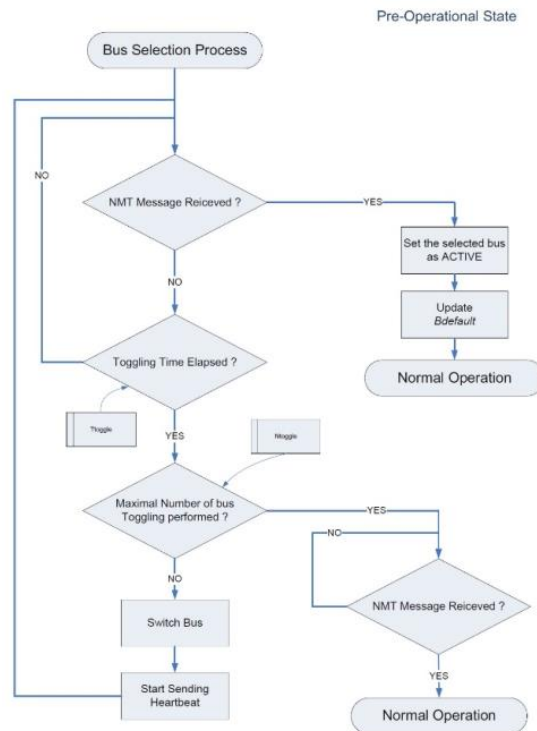
4. Ntoggle
5. Ctoggle
6. Bdefault

1. Consumer Heartbeat Time
2. Bdefault
3. Toggle

منتظر پیام ضربان قلب مدیر می‌ماند و تا زمان دریافت آن، تنها شنونده گذرگاه خواهد بود.



شکل ۳- عملکرد عادی شبکه [۹]



شکل ۲- پروسه انتخاب گذرگاه [۹]

۴ ساختار فریم CAN مبتنی بر LDUT

در ماهواره‌ها معمولاً طول داده واحدهای مختلف از حداکثر طول بخش داده فریم CAN (بایت) تجاوز می‌کند. بدین منظور، برای انتقال داده‌های بزرگ‌تر از ۸ بایت، از پروتکل انتقال واحدهای داده بزرگ (LDUT) استفاده می‌شود. قالب فریم LDUT (شکل ۴) در واقع همان ۲۹ بیت شناسه داور^۱ است. در ادامه به تشریح بخش‌های مختلف این فریم پرداخته می‌شود.

- شناسه تابع^۲: با ارزش‌ترین بخش قالب فریم، شناسه تابع است. با استفاده از شناسه تابع می‌توان داده‌های منتقل شونده بر روی گذرگاه را اولویت‌بندی نمود. باتوجه به آن که داده‌های مدیریت افزونگی گذرگاه از بیشترین اولویت برخوردارند، مقدار شناسه تابع برای داده‌های مدیریت افزونگی، بیشترین اولویت را دارند. از این رو، شناسه تابع برای داده‌های مدیریت افزونگی، کوچک‌ترین مقدار ممکن (بیشترین اولویت) را به خود اختصاص می‌دهند.

۳.۳ عملکرد عادی

در هنگام عملکرد عادی شبکه و گره، گره همواره در مدت زمان مشخصی (زمان ضربان قلب مصرف‌کننده) منتظر پیام ضربان قلب مدیر گذرگاه می‌ماند. در صورتی که در مدت زمان تعیین شده، ضربان قلب مدیر دریافت نشود، گره باتوجه به اینکه تعداد دفعات تغییر گذرگاه آن از حد مجاز گذشته است یا نه تصمیمات متفاوتی می‌گیرد. شکل ۳ روند عملکرد عادی شبکه را نشان می‌دهد. در صورتی که تعداد دفعات مجازی که می‌تواند گره گذرگاه خود را تغییر دهد (Ntoggle) به پایان رسیده باشد، گره بر روی همان گذرگاه فعلی خود باقی می‌ماند و منتظر پیام ضربان قلب مدیر می‌ماند. در این وضعیت، گره اجازه ارسال پیام بر روی گذرگاه را ندارد و تنها شنونده گذرگاه خواهد بود. در صورتی که گره اجازه تغییر گذرگاه را داشته باشد، گذرگاه خود را تغییر داده و بر روی گذرگاه جدید پیام ضربان قلب خود را ارسال می‌کند تا مدیر گذرگاه از تغییر گذرگاه آن گره مطلع شود. پس از تغییر گذرگاه، گره وارد پروسه انتخاب گذرگاه می‌شود و تمامی مراحل پروسه انتخاب گذرگاه تکرار می‌گردد.

- فریم لغو: اگر واحدی قصد لغو ارسال داده‌های قسمت‌شده را دارد، برای فرستنده فریمی با مقدار نوع فریم بخصوص را ارسال می‌نماید.
- برای ارسال داده‌های قسمت نشده، برای ارسال آخرین بخش داده‌های قسمت‌شده، برای ارسال قسمت‌های میانی داده‌های قسمت‌شده و برای ارسال اولین قسمت داده‌های قسمت‌شده از فریمی با مقدار نوع فریم متفاوت نیز استفاده می‌شود.
- آدرس مبدأ: ۷ بیت این بخش آدرس فرستنده داده را مشخص می‌کند. گیرنده می‌تواند با استفاده از این بخش میان داده‌های ارسال فرستنده‌های مختلف تمییز قائل شود.
- آدرس مقصد: ۷ بیت این بخش آدرس گیرنده داده را مشخص می‌کند. فرستنده می‌تواند با استفاده از این بخش، گیرنده‌ی داده را مشخص کند.
- شناسه پرتکل^۹: این بخش برای فریم‌های تله‌کامند/تله‌متری حاوی ۴ بیت کم ارزش شمارنده قسمت‌های داده است. برای فریم‌های دیگر مقدار این بخش متفاوت می‌باشد.
- بیت ضامن^{۱۰}
- این بیت به گیرنده کمک می‌کند تا بتواند قسمت‌های از دست رفته و یا دوباره ارسال شده را تشخیص دهد. این بیت باید به ازای هر بار ارسال فریم قسمت‌شده، خلاف (عکس) مقدار قبلی خود باشد.

بیت ضامن	شناسه پرتکل	آدرس مقصد	بدون استفاده	آدرس مبدأ	بدون استفاده	نوع فریم	شناسه تبع
۰	۴-۱	۱۱-۵	۱۲	۱۹-۱۳	۲۰	۲۳-۲۱	۲۴
۲۸							

شکل ۴- قالب فریم LDUT الگوریتم انتقال داده با استفاده از پروتکل LDUT

- به‌منظور انتقال داده با استفاده از پروتکل LDUT به روش زیر عمل می‌کنیم.
- ارسال داده:**
- (۱) ارسال فریم درخواست برای واحد OBC
 - (۲) انتظار به میزان TBD برای دریافت فریم اجازه از واحد OBC
 - (۳) دریافت فریم اجازه از واحد OBC
 - (۴) در صورت عدم دریافت فریم اجازه، مراحل ۱ تا ۳ چند بار تکرار می‌شود.

- اولویت^۱: با استفاده از این بخش می‌توان دو نوع ارسال پیشنهاد و غیرپیشنهاد را برای انتقال داده مورد استفاده قرار داد. ارسال پیشنهاد برای داده ضربان قلب مدیر گذرگاه^۲، فریم اجازه^۳، فریم لغو^۴، فریم توقف^۵، فریم ادامه^۶ و فریم تأیید نهایی^۷ مورد استفاده قرار می‌گیرد.
- نوع فریم^۸: بخش نوع فریم این امکان را فراهم می‌آورد تا بتوان انواع مختلف فریم‌هایی را که در پرتکل LDUT مورد استفاده قرار می‌گیرند از یکدیگر تمییز داد. این بخش اجازه می‌دهد تا داده‌های قسمت‌شده و قسمت نشده با کنترل دقیق‌تری ارسال شوند. فریم‌های LDUT با توجه به مقداری که در این بخش مشخص می‌شود، در ۸ نوع دسته متنوع قرار می‌گیرند.
 - فریم توقف: فریم توقف زمانی تولید می‌شود که واحد گیرنده (مقصد) خطایی در ترتیب فریم‌های دریافتی تشخیص دهد و یا حافظه آن پر شده باشد. گره‌ای که فریم توقف را ارسال کرده است، باید پس از دریافت تأیید آن، فریم ادامه را ارسال کند.
 - فریم تأیید: هر واحد باید به ازای هر داده‌ی کامل و صحیحی که دریافت می‌کند، فریم تأیید آن را برای فرستنده ارسال نماید. این تأیید شامل فریم‌هایی می‌شود که حاوی داده‌های قسمت‌شده است. توجه شود که فریم تأیید تنها هنگامی ارسال می‌شود که داده به‌صورت کامل دریافت شده باشد. بدین معنی که برای داده‌های قسمت‌شده فریم تأیید پس از دریافت آخرین فریم قسمت‌شده داده ارسال می‌شود. نکته مهم در تأیید داده‌های دریافتی آن است که تمامی داده‌هایی که به‌صورت قسمت‌شده دریافت می‌شوند باید تأیید شوند. در پیاده‌سازی مورد نظر، فرض بر آن است که تمامی داده‌های تله‌کامند/تله‌متری پس از دریافت توسط مقصد باید تأیید شوند. نوع دیگری از فریم تأیید برای تأیید دریافت فریم توقف استفاده می‌شود. تفاوت این دو فریم در مقدار بخش اولویت است.
 - فریم ادامه: واحدی که فرمان توقف در ارسال را داده است، برای راه‌اندازی مجدد واحد متوقف‌شده باید ادامه فریم را ارسال کند.

6. Resume Frame
7. Acknowledgment (ACK)
8. Frame Type
9. Protocol ID
10. Toggle Bit

1. Priority
2. Network Management
3. Permission Frame
4. Abort Frame
5. Stop Frame

گرفتن سخت افزار عملگر چرخ عکس‌عملی مرتبط با بخش ارتباطی CAN و همچنین مدیریت افزونگی در مواجهه با خطاهای احتمالی سخت افزاری و نرم افزاری بررسی می‌گردد. از قطعات زیر در پیاده‌سازی مرتبط با این قرارداد در سخت افزار این عملگر استفاده شده‌است:

PIC32MZ Embedded Connectivity
Communication Interfaces
•Two CAN modules (with dedicated DMA channels):
-2.0B Active with DeviceNet™ addressing support

و مبدل CAN با مشخصه زیر

LTM2889
Isolated CAN FD μModule
Transceiver and Power

در حالت ارتباط عادی چرخ‌های عکس‌عملی با واحد OBC از طریق گذرگاه CAN فعال، فرمان‌های موردنظر در قالب یک فریم حاوی سه بایت داده به چرخ مربوطه گسیل می‌شوند. در این بخش تنها به پروتکل و فرمت فیلدهای حاوی کدها و مقادیر فرمان چرخ‌های عکس‌عملی پرداخته می‌شود و بقیه فیلدهای فریم‌های CAN در مطابقت با ساختار فریم CAN و مدیریت افزونگی گذرگاه توضیح داده شده خواهد بود. برای مثال، بخشی از فریم که در برگزیده بیت‌های شناسایی و داوری می‌باشد به شکل ۵ خواهد بود:

Identifier		SRR	IDE	Extended Identifier				RTR	RB1	RBO
Bits (10:2)	Bits (1:0)			Bits (17:13)	Bit12	Bits (11:5)	Bits (4:0)			
0x29	SAD (6:5)	1	1	SAD (4:0)	1	DAD (6:0)	0x1F	0	0	0

شکل ۵- بخشی از ساختار فریم CAN در برگزیده بیت‌های شناسایی و داوری

در ارسال فرامین به چرخ‌های عکس‌عملی مقدار SAD برابر با آدرس واحد OBC و مقدار DAD برابر با آدرس چرخ عکس‌عملی مورد نظر می‌باشد و در هنگام دریافت مقادیر تله‌متری، بالعکس. اطلاعات تله‌متری چرخ‌ها نیز در قالب دو فریم پیاپی ۸ بیتی به واحد OBC برگردانده می‌شوند که در ادامه شرح داده شده‌است.

۵.۱ نحوه ارسال فرامین به چرخ‌های عکس‌عملی

فرمان‌های اعمالی به چرخ‌های عکس‌عملی در قالب یک فریم CAN حاوی سه بایت داده روی گذرگاه ارسال می‌شوند. بایت سوم (با ارزشترین بایت - دارای تقدم زمانی) از فیلد داده‌ها مربوط به کد فرمان و دو بایت بعدی مربوط به مقدار فرمان می‌باشد. فیلد طول داده (DLC) در ارتباط CAN برای فرامین اعمالی به موتور همواره برابر مقدار ۳ خواهد بود. همانگونه که ذکر شد، در شکل ۶ تنها فیلدهای مربوط به داده‌ها در فریم CAN نمایش داده شده‌اند.

- (۵) ارسال اولین فریم داده، سپس فریم‌های میانی داده و در نهایت آخرین فریم داده به واحد OBC
- (۶) انتظار به میزان TBD برای دریافت فریم تأیید نهایی از واحد OBC
- (۷) دریافت فریم تأیید نهایی از واحد OBC
- (۸) در صورت عدم دریافت فریم تأیید نهایی، کلیه مراحل سه بار تکرار می‌شود.

چنانچه در هنگام ارسال داده فریم توقف دریافت شود:

- (۱) ارسال فریم تأیید توقف
- (۲) انتظار به میزان TBD برای دریافت فریم ادامه
- (۳) دریافت فریم ادامه
- (۴) در صورت عدم دریافت فریم ادامه، ارسال داده از سر گرفته می‌شود (ارسال فریم درخواست و...)
- (۵) ادامه ارسال فریم‌های قسمت شده از شماره قسمتی که در بایت اول داده فریم ادامه به آن اشاره شده‌است.

دریافت داده:

- (۱) دریافت فریم درخواست
 - (۲) ارسال فریم اجازه
 - (۳) دریافت اولین فریم داده، سپس فریم‌های میانی داده و در نهایت آخرین فریم داده
 - (۴) انتظار به میزان TBD برای دریافت هر قسمت از داده
 - (۵) در صورت عدم دریافت داده در مدت زمان مشخص، پرتکل از حالت دریافت داده خارج می‌شود.
 - (۶) ارسال فریم تأیید نهایی
- چنانچه در هنگام دریافت داده، شماره شناسه پروتکل فریم دریافتی با شماره داده قسمت شده مورد انتظار یکسان نباشد فریم توقف ارسال می‌شود:
- (۱) انتظار به میزان TBD ثانیه برای دریافت فریم توقف
 - (۲) دریافت فریم تأیید فریم توقف
 - (۳) در صورت عدم دریافت تأیید فریم توقف، پروتکل از حالت دریافت داده خارج می‌شود.
 - (۴) ارسال فریم ادامه با محتوی یک بایتی
 - (۵) ادامه دریافت فریم‌های قسمت شده

۵ پیاده‌سازی و ارزیابی قرارداد LDUT بر روی

عملگر چرخ عکس‌عملی

در این قسمت، قرارداد LDUT روی عملگر چرخ عکس‌عملی پیاده‌سازی و برای کنترل نحوه ارسال / دریافت فرامین مختلف با در نظر

به صورت زمانی) و ... می‌باشد. بنابراین صفر بودن بایت وضعیت به مفهوم عملکرد طبیعی و بدون مشکل چرخ عکس‌العملی است. در شکل‌های ۱۰ و ۱۱ بستر تست لازم به صورت تکی و خروجی که برچسب گذاری شده، به نمایش درآمده است. کدهای فرمان در نظر گرفته شده شامل فرمان‌های سرعت/گشتاور در جهت‌های چپگرد، راستگرد (۲۴۱ / ۲۵۱) با یک اختلاف برای چپگرد و راستگرد می‌باشند. دو فرمان حالت استندبای و استعلام مقادیر تله‌متری نیز با کدهای ۲۳۱ و ۲۲۰، و ۰ و ۲۲۰ به این مجموعه کدها اضافه گردیده‌اند. توضیح اینکه بایت‌های متعاقب مربوط به فرمان استندبای و استعلام تله‌متری برابر با مقدار صفر ارسال می‌شوند.

ارزش عددی (رزولوشن) مقدار فرمان سرعت برابر rpm ۱ و ارزش عددی مقدار فرمان گشتاور برابر با ۰/۱ mNm لحاظ می‌شود. بنابراین مقدار rpm ارسال شده با دستور تعیین سرعت برای یک چرخ برابر خواهد بود با:

$$\text{Speed} = (256 \times \text{MSB} + \text{LSB}) \times 1 \text{ (rpm)}$$

و مقدار دستور گشتاور (با واحد mNm) اعمالی به چرخ عکس‌العملی برابر خواهد بود با:

$$\text{Trq} = (256 \times \text{MSB} + \text{LSB}) \times 0.01 \text{ (mNm)}$$



شکل ۱۰ - بستر تست و صحنه‌سنجی توسط آنالایزر CAN

0x200120F x	0	0x1	0x2	0x16	0x3	0x20	0x0	0x90	0x00
0x200120F x	3	0xF4	0x01	0xF1	0x03	0x20	0x00	0x90	0x00
0x000 x	4	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x200120F x	3	0x00	0x00	0xDC	0x00	0x00	0x00	0x00	0x00
0x1120030F x	1	0x01	0x00	0xDC	0x00	0x00	0x00	0x00	0x00
0x1090320F x	0	0x02	0x00	0xDC	0x00	0x00	0x00	0x00	0x00
0x3F28011 x	8	0x00	0x03	0x00	0xF1	0x00	0x3C	0xF8	0x00
0x3082022 x	8	0x1	0x02	0x16	0x03	0x1E	0x00	0x93	0x00
0x200120F x	0	0x1	0x02	0x16	0x03	0x1E	0x00	0x93	0x00
0x200120F x	3	0xF4	0x01	0xF1	0x03	0x1E	0x00	0x93	0x00
0x000 x	8	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x200120F x	3	0x00	0x00	0xDC	0x00	0x00	0x00	0x00	0x00
0x1120030F x	1	0x01	0x00	0xDC	0x00	0x00	0x00	0x00	0x00
0x200120F x	0	0x01	0x00	0xDC	0x00	0x00	0x00	0x00	0x00
0x3F28021 x	8	0x00	0x14	0x00	0xF1	0xF8	0x3C	0xF8	0x00
0x3082022 x	8	0x1	0x02	0x16	0x03	0x1E	0x00	0x93	0x00
0x200120F x	0	0x1	0x02	0x16	0x03	0x1E	0x00	0x93	0x00
0x200120F x	3	0xF4	0x01	0xF1	0x03	0x1E	0x00	0x93	0x00
0x000 x	8	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x200120F x	3	0x00	0x00	0xDC	0x00	0x00	0x00	0x00	0x00
0x1120030F x	1	0x01	0x00	0xDC	0x00	0x00	0x00	0x00	0x00
0x000 x	8	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x200120F x	1	0x01	0x00	0xDC	0x00	0x00	0x00	0x00	0x00
0x000 x	8	0x00	0x00	0xDC	0x00	0x00	0x00	0x00	0x00
0x200120F x	0	0x01	0x00	0xDC	0x00	0x00	0x00	0x00	0x00
0x3F28021 x	8	0x00	0x03	0x00	0xF2	0xF0	0x3C	0xF8	0x00
0x3082022 x	8	0x0F	0x02	0x16	0x03	0x1D	0x00	0x90	0x00
0x200120F x	0	0x0F	0x02	0x16	0x03	0x1D	0x00	0x90	0x00
0x200120F x	3	0x00	0x00	0xF7	0x03	0x1D	0x00	0x90	0x00

شکل ۱۱ - خروجی صحنه‌گذاری شده پروتکل CAN مجزا

DLC	Data Byte 2	Data Byte 1	Data Byte 0
۳	کد فرمان	مقدار (MSB)	مقدار (LSB)

شکل ۶ - فرمان‌های اعمالی به چرخ‌های عکس‌العملی در قالب فریم CAN

۲.۵ فرمت و طریقه دریافت اطلاعات تله‌متری چرخ‌ها روی گذرگاه CAN

هر چرخ پس از دریافت فرمان استعلام مقادیر تله‌متری از طریق گذرگاه CAN، دو فریم را با آدرس فرستنده مربوط به خود و آدرس گیرنده واحد OBC روی گذرگاه گسیل می‌نماید. فریم اول (دارای تقدم زمانی) که دارای طول داده هفت بیتی می‌باشد حاوی جهت و مقادیر سرعت اندازه‌گیری شده و شتاب و همچنین اعلام وضعیت موتور است. مقادیر کمیت‌ها در دو بایت و کد مشخصه جهت سرعت و شتاب و همچنین سیگنال وضعیت هر کدام در یک بایت قرار می‌گیرند. فریم دوم که بلافاصله بعد از فریم اول ارسال می‌شود، هشت بیتی بوده و در بر دارنده اطلاعات مربوط به وضعیت کارکرد موتور (مقادیر ولتاژ، جریان و دماهای اندازه‌گیری شده) می‌باشد. در اینجا نیز مقدار هر کمیت در دو بایت قرار می‌گیرد؛ بنابراین هر موتور با دریافت یک فریم CAN حامل فرمان که در شکل ۷ آورده شده است.

DLC	Data Byte 2	Data Byte 1	Data Byte 0
۳	۲۲۰	۰	۰

شکل ۷ - یک فریم CAN حامل فرمان دریافتی در موتور

دو فریم پاسخ تله‌متری را به دو شکل ۸ و ۹ ارسال می‌نماید:

DLC	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
۸	جهت شتاب		سرعت		جهت شتاب		وضعیت	
	سرعت							

شکل ۸ - پاسخ تله‌متری موتور

DLC	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
۸	ولتاژ		جریان		دمای برد		دمای موتور	
							1	

شکل ۹ - پاسخ تله‌متری موتور

بایت وضعیت^۱ نمایاگر وجود خطا در عملکرد بخش‌هایی از چرخ می‌باشد. ارزش "1" هر بیت از این بایت متناظر با وجود یا عدم وجود خطا (بروز Overload، خرابی گذرگاه CAN (تشخیص ندادن فریم‌ها

1. Status byte

- دمای موتور برابر ۶۰ درجه سانتیگراد می‌باشد:

$$(4 \times 256 + 76 - 500) \times 0.1 = 60$$

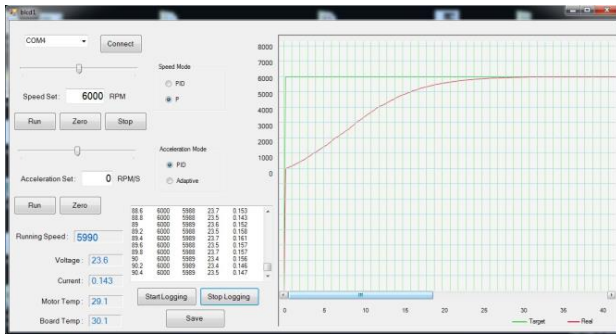
- دمای درایور برابر ۵۲/۴ درجه سانتیگراد می‌باشد:

$$(4 \times 256 + 0 - 500) \times 0.1 = 52.4$$

در ادامه شکل‌های ۱۵ و ۱۶ برای مجموعه ستاپ و نمودار خروجی عملکرد دستگاه تحت تست (چرخ عکس‌العملی) اضافه گردیده است. باتوجه به شکل ۱۵، مجموعه ستاپ متشکل از چرخ عکس‌العملی، منبع تغذیه، شبیه‌ساز پروتکل CAN و آنالایزر گذرگاه CAN به همراه لپ تاپ برای اجرای و دریافت دستورات می‌باشد. در شکل ۱۶ نیز دستور سرعت هدف و سرعت رسیده شده به همراه اطلاعات تله‌متری در GUI پیاده‌سازی شده آورده شده است.



شکل ۱۵- مجموعه ست آپ تست



شکل ۱۶- نمودار خروجی اجرای دستور سرعت ۶۰۰۰ دور در دقیقه و رصد اطلاعات تله‌متری- ۱۰ هرتز

۳.۵ مدیریت افزونگی

در این بستر تست برای اولین بار قرارداد LDUT طبق استاندارد فضایی اروپا ECSS بر روی عملگرهای کنترل وضعیت یک ماهواره فرضی بررسی و بطور کامل پیاده‌سازی شده است. مبحث خطا را نیز به دو صورت سخت افزاری و نرم افزاری می‌توان در نظر گرفت. در بحث سخت افزاری، چنانچه هر یک از سیم‌های (اصلی و افزونه L_CAN) و H_CAN به Ground متصل و یا اتصال یکی از دو سیم قطع شود، طبق پروتکل مندرج و الگوریتم شکل ۲ به باس دیگر سوییچ خواهد کرد

به عنوان مثال، ارسال فرمان سرعت راستگرد rpm ۳۰۰۰ را در نظر می‌گیریم. فریم ارسالی روی باس باید حاوی فیلدهایی با مقادیر زیر باشد:

$$(11 \times 256 + 184) \times 1 = 3000$$

DLC	Data Byte 2	Data Byte 1	Data Byte 0
۳	۲۵۱	۱۱	۱۸۴

شکل ۱۲- نحوه ارسال فرمان سرعت راستگرد rpm ۳۰۰۰ به موتور

در خصوص ارزش عددی مقادیر اندازه‌گیری شده تله‌متری (دو بیتی) توجه به این نکات ضروری است:

- سرعت با رزولوشن ۱ rpm ارسال می‌شود.
- شتاب با رزولوشن ۱ rpm/s ارسال می‌گردد.
- ولتاژ با رزولوشن ۰/۱ ولت ارسال می‌شود.
- جریان با رزولوشن ۱۰ میلی آمپر ارسال می‌گردد.
- دمای موتور و دمای برد درایور هر کدام با رزولوشن ۰/۱ درجه سانتیگراد و با مقدار آفست ۵۰۰ ارسال میشوند. وجود مقدار آفست ذکر شده سبب می‌گردد که امکان ارسال مقادیر منفی نیز در قالب یک عدد مثبت فراهم شود. برای مثال، با در نظر گرفتن فریم‌های ارسالی از جانب موتور مطابق شکل‌های ۱۳ و ۱۴:

DLC	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
A	۰	۲۵۲	۵	۱۲۰	۲۴۱	۰	۶۰	۰
	جهت سرعت		سرعت		جهت شتاب		شتاب	

شکل ۱۳- فریم ارسالی از جانب موتور

DLC	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
A	۱	۲۴	۰	۵۱	۴	۷۶	۴	۰
	ولتاژ		جریان		دمای موتور		دمای برد	

شکل ۱۴- فریم ارسالی از جانب موتور

مقادیر تله‌متری بگونه زیر خواهد بود:

- سرعت چرخ برابر ۱۴۰۰ دور بر دقیقه در جهت چپگرد می‌باشد:

$$(5 \times 256 + 120) \times 1 = 1400$$

- شتاب چرخ برابر ۶۰ rpm بر ثانیه در جهت راستگرد می‌باشد:

$$(0 \times 256 + 60) \times 1 = 30$$

- چرخ بدون خطا کار می‌کند:

$$\text{Status} = 0$$

- ولتاژ موتور برابر ۲۸ ولت است:

$$(1 \times 256 + 24) \times 0.1 = 28$$

- جریان موتور برابر ۵۱۰ میلی آمپر است:

$$(0 \times 256 + 51) \times 0.01 = 510 \text{ mA}$$

جنبه‌های مورد نیاز برای برآوردن نیازهای خاص سیستم‌های مدیریت داده ماهواره‌ها را پوشش دهد. استاندارد مذکور یکی از سری استانداردهای ECSS است که به رابط‌های پیوند داده و پروتکل‌های ارتباطی مربوط می‌شود. همه موارد مطرح شده در عملگرهای کنترل وضعیت چرخ عکس‌العملی از لحاظ سخت افزاری و نرم افزاری پایه‌سازی و توسط آنالایزر CAN و شبیه ساز OBC مورد صحت‌گذاری قرار گرفت. در ادامه مدیریت افزونگی و فرآیند انتخاب گذرگاه بررسی و ساختار فریم CAN مبتنی بر LDUT با تاکید بر الگوریتم انتقال داده بررسی و پیشنهاد گردید.

تعارض منافع

هیچ‌گونه تعارض منافع توسط نویسندگان بیان نشده‌است

مراجع

- [1] A. Scholz, J. N. Juang, P. Mader, J. Schlegel, and M. Starcik, "Space CAN-A low-cost, reliable and robust control and monitoring bus for small satellites," *Acta Astronautica*, vol. 161, pp. 1-11, 2019, <https://doi.org/10.1016/j.actaastro.2019.05.010>.
- [2] D. Bracknell, "Introduction to the Mil-Std-1553B serial multiplex data bus," *Microprocessors and Microsystems*, vol. 12, no. 1, pp. 3-12, 1988, [https://doi.org/10.1016/0141-9331\(88\)90030-0](https://doi.org/10.1016/0141-9331(88)90030-0).
- [3] D. Roberts and S. Parkes, "Spacewire missions and applications," in *International SpaceWire International SpaceWire Conference*, Saint Petersburg, Russia, 2010, pp. 431-436.
- [4] J. Puig-Suari, C. Turner and W. Ahlgren, "Development of the standard CubeSat deployer and a CubeSat class PicoSatellite," in *IEEE Aerospace Conference Proceedings (Cat. No.01TH8542)*, Big Sky, MT, USA, 2001, pp. 1-7, <https://doi.org/10.1109/AERO.2001.931726>.
- [5] M. Noca et al., "Lessons learned from the first Swiss pico-satellite: SwissCube," in *23rd Annual AIAA/USU Conference on Small Satellites*, Logan, Utah, USA, 2009, Paper SSC09-XII-9.
- [6] H. J. Kramer and A. P. Cracknell, "An overview of small satellites in remote sensing," *International journal of remote Sensing*, vol. 29, no. 15, pp. 4285-4337, 2008, <https://doi.org/10.1080/01431160801914952>.
- [7] G. Casarosa, M. Apuzzo, L. Fanucci, and B. Sarti, "Characterization of the EMC performances of the can bus in a typical system bus architecture for small satellites," in *9th EUROMICRO Conference on Digital System Design (DSD'06)*, Cavtat, Croatia,

و این عمل هم بصورت تکی و هم به صورت تجمیع بر روی باس CAN ماهواره تست و با موفقیت این جایگزینی باس انجام یافته است. در مورد تست نرم افزاری نیز می توان بیان نمود که برای مثال در Checksum طوری در برنامه هر یک از عملگرها تغییر داد که بتواند بعد از گذشت یک مدت‌زمان معین این جایگزینی (برای مثال هر ۵ دقیقه) در باس انجام شود که این نیز با موفقیت تست شده‌است. همچنین می توان از طریق Error Count های برنامه وقتی به تعداد معینی برسد باز این سویچ انجام گیرد که این مورد هم با موفقیت انجام یافته است.

۴.۵ بحث و بررسی

در پایه‌سازی این پروتکل بر روی هر یک از گره‌های گذرگاه ماهواره، نیاز به سیستمی وجود دارد که از لحاظ ارسال و دریافت و مدیریت گذرگاه (OBC) برای سویچ و فرآیند نظارت شبیه‌سازی شده باشد. نرخ انتقال داده‌برداری در گره مورد نظر بر روی ۵۰۰ kbps تنظیم گردیده و محل نمونه برداری بین ۸۰ تا ۹۰ درصد طول بیت بوده و همچنین در ساختار مداری چرخ عکس‌العملی نیازی به موتناژ مقاومت termination (مقاومت ۱۲۰ اهمی) نبوده است. سویچ بین دو گذرگاه CAN توسط هر گره به این صورت انجام گرفت که پس از هربار سویچ CAN روی باس فعال، فریم Bootup نیز اضافه گردید.

عدم اجرای فرمان توسط چرخ عکس‌العملی نیز در صورت نبود تأییدیه فریم NMT روی گذرگاه توسط شبیه ساز OBC ماهواره با موفقیت تست گردید. ترتیب بایت‌های مربوط به داده‌ها و فرامین چرخ عکس‌العملی براساس MSB (بایت پرارزش) بودن بایت‌های با تقدم زمانی در سمت OBC بوده و فریم‌های توقف و ادامه نیز به‌منظور بازیابی خرابی انتقال داده مطابق پروتکل LDUT انجام گردید. تعیین وضعیت سلامتی یا خطای هریک از بیت‌های مربوط به وضعیت چرخ عکس‌العملی در هر نمونه^۱ زمانی به‌صورت مستقل از سمپل‌های زمانی پیشین، پایه‌سازی شده بود. پایه‌سازی همزمان هر چهار گره (چرخ عکس‌العملی) در کنار گره‌های دیگر ماهواره بدون تداخل با موفقیت انجام گرفت.

۶ نتیجه گیری

در این مقاله، اهمیت پروتکل ارتباطی CAN در کاربردهای فضایی و الزامات تشخیص خطا در گذرگاه بیان گردید. همچنین به الزامات استاندارد استفاده از گذرگاه داده CAN برای کاربردهای فضایی پرداخته شد. این الزامات مشخصات شبکه CAN را گسترش می‌دهد تا

- [10] O. Pfeiffer, A. Ayre, and C. Keydel, *Embedded Networking with CAN and CANopen*, Greenfield, MA, USA :Copperhill Technologies Corporation ,2008.
- [11] E. Secretariat, "CANBus extension protocol," 2013, <https://doi.org/10.3403/30340718u>.
- [12] S. Wielandt, G. Ottoy, J. P. Goemaere, N. Stevens, and L. De Strycker, "Integration of a CAN bus in an onboard computer for space applications," in 11th International Conference on Development and Application Systems, Suceava, Romania, 2012, pp. 56-59.
- 2006, pp. 338-345, <https://doi.org/10.1109/DSD.2006.99>.
- [8] M. Khurram and S. M. Y. Zaidi, "CAN as a spacecraft communication bus in LEO satellite mission," in *2nd International Conference on Recent Advances in Space Technologies*, Istanbul, Turkey, 2005, pp. 432-437, <https://doi.org/10.1109/RAST.2005.1512606>.
- [9] K. Janschek and A. Braune, "Application of industrial CAN bus technology for LEO-satellites," *Acta Astronautica*, vol. 46, no. 9-2, pp. 313-317, 2000, [https://doi.org/10.1016/S0094-5765\(99\)00221-0](https://doi.org/10.1016/S0094-5765(99)00221-0).