Available in:
Journal.isrc.ac.ir

Journal of Space Science, Technology & Applications (Persian)

Vol. 2, No. 2, pp.: 149-157
2022

DOI:
10.22034/jssta.2023.374476.1102

Article Info

Received: 2022-12-3
Accepted: 2023-2-7

Keywords

Satellite, Onboard software, Test results, Lesson learned

How to Cite this article

Fateme Salarkaleji , Parvin shavandi, Alireza Omranian, Oveys Kazemi Farzad Emami ,Shahrokh jalilian , Alireza khani, Abolfazl Dayyani, Mohammad Sayanjali. "Lessons Learned and Achievements in Pars1 Satellite Onboard Software Development and Test". *Journal of Space Science, Technology and Applications*, vol 2 (2), p.: 149-157, 2023.

Original Article

Lessons Learned and Achievements in Pars1 Satellite Onboard Software Development and Test

Fatemeh Salarkaleji[1*], Parvin Shavandi[1], Alireza Omranian[1], Oveys Kazemi[1], Farzad Emami[1] Shahrokh Jalilian[1], Alireza khani[1], Abolfazl Dayyani[1], Mohammad Sayanjali[1]

1. Power and Data Processsing Research Group, Satellite Research Institute,Tehran, Iran
* Corresponding Author fatemeh.salar@gmail.com

Abstract

PARS1 Satellite is a Remote Sensing satellite with a 3-years mission. The main mission of this satellite is imaging the earth by three cameras named MS, SWIR and TIR. PARS1 OnBoard Software (OBSW) is developed as a performance platform, satellite components control, manage their data, algorithm management included normal status control and event handling. OBSW is more important than other satellite subsystems due to its complexity and different features. So, the design, development and test of the PARS1 satellite OBSW is a useful platform to gain valuable experiences in the field of satellite onboard software which is very wide and complicated in its field. Therefore, in this paper decided to present experiences which gained in this field, in the form of summary of achievements and lessons learned. These experiences is gained from development and test phase of the satellite and using of these experiences will be very useful and effective in smoothing of test and development path in future projects of the Satellite Research Institute

مقاله پژوهشی

دسترس‌پذیر در نشانی:
Journal.isrc.ac.ir

دو فصلنامه
علـوم، فـناوری و
کاربردهـای فضـایی

سال دوم، شماره ۲، صفحه ۱۵۷-۱۴۹
پاییز و زمستان ۱۴۰۱

DOI:
10.22034/jssta.2023.374476.1102

درس آموخته‌ها و دستاوردهای کسب شده در توسعه و تست نرم‌افزار روی‌برد ماهواره پارس ۱

فاطمه سالارکالجی*[۱]، پروین شوندی[۱]، علیرضا عمرانیان[۱] ، اویس کاظمی[۱] ، فرزاد امامی[۱]، شاهرخ جلیلیان[۱]، علیرضا خانی[۱]، ابوالفضل دیانی[۱]، محمد سینجلی[۱]

۱. گروه پژوهشی توان و پردازش داده، پژوهشکده سامانه های ماهواره، تهران، ایران
*. نویسنده مسئول fatemeh.salar@gmail.com

چکیده

ماهواره پارس یک، یک ماهواره سنجشی با مأموریت سه ساله است. مأموریت اصلی این ماهواره، تصویربرداری از زمین توسط سه دوربین MS، SWIR و TIR است. نرم‌افزار ماهواره پارس یک به عنوان بستر اجرای سناریو ماهواره، بررسی سلامت اجزای مختلف ماهواره، مدیریت داده‌های آن‌ها و اجرای الگوریتم‌های زیرسیستمی شامل کنترل حالت‌های عادی و رخدادهای نامناسب، طراحی و پیاده‌سازی شده است. نرم‌افزار روی‌برد به دلیل پیچیدگی و ویژگی‌های متفاوتش نسبت به دیگر زیرسیستم‌های ماهواره، از اهمیت بالاتری برخوردار است. طراحی، توسعه و تست نرم‌افزار روی‌برد ماهواره پارس یک، بستر مفیدی برای کسب تجارب ارزشمندی برای بخش نرم‌افزار روی‌برد ماهواره بوده که دارای گستردگی زیادی است. از این‌رو، در این مقاله تصمیم گرفته شد، تجربه‌های کسب شده در این زمینه، در قالب خلاصه‌ای از دستاوردها و درس آموخته‌ها ارائه شود. تجارب کسب شده حاصل از فاز توسعه و تست ماهواره است و استفاده از این تجربیات و دستاوردها در هموار کردن مسیر توسعه و تست پروژه‌های آینده پژوهشکده ماهواره بسیار مفید و مؤثر خواهد بود.

تاریخچه داوری

دریافت: ۱۴۰۱/۰۹/۱۲

پذیرش: ۱۴۰۱/۱۱/۱۸

واژه‌های کلیدی

ماهواره، پارس ۱، نرم نرم‌افزار روی برد، تست، دستاوردها، درس آموخته.

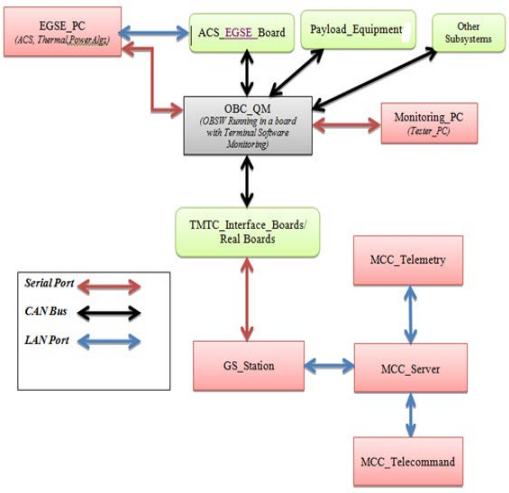
نحوه استناد به این مقاله

فاطمه سالارکالجی، پروین شوندی، علیرضا عمرانیان ، اویس کاظمی ، فرزاد امامی، شاهرخ جلیلیان، علیرضا خانی، ابوالفضل دیانی، محمد سینجلی. "درس آموخته‌ها و دستاوردهای کسب شده در توسعه و تست نرم‌افزار روی‌برد ماهواره پارس ۱." *دوفصلنامه علوم، فناوری و کاربردهای فضایی*، جلد دوم، شماره دوم، صفحات ۱۵۷-۱۴۹، ۱۴۰۱.

Abbreviations	
MS	MultiSpectral
SWIR	Short Wave Infrared
TIR	Thermal Infrared
MOP	Mission Operation Plan
FDIR	Fault Detection Isolation and Recovery
ECSS-PUS	European Cooperation for Space Standardization Packet Utilization Standard-Packet Utilization Standard
CAN	Controller Area Network
OBC	On Board Computer
FD	Fault Detection
HK Data	Housekeeping Data
PIL	Process- In-the -Loop
SID	Structure Identification
MRAM	Magneto resistive Random Access Memory

۱-مقدمه

ماهواره پارس یک، یک ماهواره سنجشی با مأموریت سه ساله است. مأموریت اصلی این ماهواره، تصویربرداری از زمین توسط سه دوربین MS[1] ، SWIR[2] و TIR[3] است. نرم‌افزار ماهواره پارس یک به عنوان بستر اجرای سناریو ماهواره، بررسی سلامت اجزای مختلف ماهواره، داده‌پراکنی و داده‌برداری از آن‌ها و اجرای الگوریتم‌های زیرسیستمی شامل کنترل حالت‌های عادی و رخدادهای نامناسب، طراحی و پیاده‌سازی شده است. با توجه به این توانمندی‌ها و مأموریت‌های ویژه‌ای که برعهده این ماهواره قرار دارد و با توجه به پیچیدگی‌ها، تعدد و اهمیت وظایف آن، ضریب اطمینان از کارکرد صحیح زیرسیستم‌ها از جمله زیرسیستم فرمان و مدیریت داده و سخت‌افزار و نرم‌افزار آن از اهمیت بالایی برخوردار است. به همین دلیل، تلاش برای طراحی و مدیریت عملکرد نرم‌افزار روی‌برد زیرسیستم مدیریت فرمان و داده به منظور جلوگیری از رخدادهای ناخوشایند و مدیریت صحیح مأموریت ماهواره، بسیار حائز اهمیت است. شکل ۱، نمایی از بستر تست مدل کیفی ماهواره پارس یک را نشان می‌دهد که به صورت شماتیک و سطح بالا به تصویر کشیده شده است.



شکل۱. شمای شماتیک بستر تست مدل کیفی ماهواره پارس یک

در طول فرایند پیاده‌سازی و تست نرم‌افزار روی‌برد ماهواره پارس یک، تجربه‌های مفید و قابل توجهی کسب شد که مستند کردن این تجربیات از ضروریات هر پروژه و همین‌طور پروژه پارس یک است. در این مقاله، به صورت مختصر، مجموعه‌ای از این تجربیات در قالب درس آموخته‌ها و دستاوردها جمع‌بندی شده است.

استفاده از این درس آموخته‌ها و تجربیات در هموار کردن مسیر توسعه و تست پروژه‌های آینده پژوهشکده سامانه‌های ماهواره تأثیرگذار خواهد بود.

به طور کلی در سطح بالا، دو نگاه به تست نرم‌افزار وجود دارد:

در نگاه اول، توسعه دهنده[4] برنامه‌نویس یا توسعه دهنده به توسعه پایه‌ای‌ترین سطح یک برنامه (فانکشن/کلاس) تا توسعه مربوط به تجمیع دو یا چند فانکشن با یکدیگر پرداخته و تست مربوط به هر بخش را بعد از اتمام برنامه‌نویسی یا توسعه در محیط تست توسعه دهنده انجام می‌دهد.

در نگاه دوم یا نگاه کاربری[5]، تحویل گیرنده نهایی محصول نرم‌افزار یا تست کننده نهایی، در محیط عملیاتی و نهایی تست را انجام می‌دهد.

1 MultiSpectral
2 Short Wave Infrared
3 Thermal Infrared

4 Developer
5 End User/ Tester

از آنجاکه پروژه ماهواره پارس یک، یک پروژه محصول‌محور بوده و باید به سازمان فضایی ایران تحویل داده می‌شد، هدف نهایی در این پروژه، تست با نگاه کاربری یا همان تحویل گیرنده نهایی بوده به طوری که، تیم تحویل گیرنده نهایی پروژه از تست‌های مربوط به درستی و صحت عملکرد ماهواره پارس یک، به صورت تجمیع شده اطمینان حاصل کند.

مهم‌ترین دستاورد این پروژه این بود که تست تجمیع شده ماهواره با اطمینان به اینکه تمامی الزامات مشخص شده در سیستم را پوشش می‌دهد، به کاربر نهایی یا تیم تحویل گیرنده محصول نهایی (ماهواره پارس یک) ارائه شد و این نتیجه تلاش و کار تیمی و نیز تعامل خوب بین این بخش و بخش سخت‌افزار کامپیوتر روی‌برد بود.

در این مقاله، ابتدا ویژگی‌های نرم‌افزار روی‌برد ماهواره پارس یک به صورت موردی بیان شده و سپس در بخش‌های بعدی، مجموعه مختصر و مفیدی از درس آموخته‌ها و دستاوردها در حین توسعه و تست نرم‌افزار روی‌برد ماهواره ارائه می‌شود.

۲- ویژگی‌های نرم‌افزار روی‌برد ماهواره پارس یک

با توجه به اهمیت موارد مطرح شده می‌توان از قابلیت‌های مهم و کلیدی نرم‌افزار روی‌برد ماهواره پارس یک به موارد زیر اشاره کرد:

- مدیریت[1] یا همان عملیات مأموریت ماهواره شامل اجرای کامل سناریوهای هر فاز و مود ماهواره
- مدیریت تشخیص خرابی[2] زیرسیستم‌های ماهواره شامل زیرسیستم توان، حرارت، کنترل وضعیت و مخابرات
- مدیریت پردازش تله‌کامند ارسالی از زمین به ماهواره مطابق با استاندارد ECSS-PUS
- مدیریت تله‌متری تولیدی از ماهواره به ایستگاه زمینی به منظور مانیتورینگ مطابق با استاندارد ECSS-PUS
- مدیریت ذخیره‌سازی و بازیابی داده‌های روی حافظه‌های ماندگار مطابق با استاندارد ECSS-PUS
- مدیریت رخدادها مطابق با استاندارد ECSS-PUS
- مدیریت اجرای الگوریتم‌ها و داده‌برداری از حسگرها و عملگرهای کنترل وضعیت
- مدیریت محموله تصویربرداری شامل تصویربرداری در مودهای زمان واقعی و شبه زمان واقعی، مود ذخیره و ارسال تصویر ذخیره شده و نیز دریافت متادیتاهای پیوست شده تصویر
- مدیریت سناریوهای از قبل زمان‌بندی شده و اجرای هر یک در برچسب زمانی مرتبط مطابق با استاندارد ECSS-PUS
- قابلیت بارگذاری نرم‌افزار به‌روز شده و دستور به‌روزرسانی آن از ایستگاه زمینی (پیاده‌سازی راه‌انداز[3]) مطابق با استاندارد ECSS-PUS
- رمزنگاری و احراز هویت تله‌کامندهای ارسال شده از ایستگاه زمینی به ماهواره در نرم‌افزار روی‌برد ماهواره
- قابلیت ماژولاریتی نرم‌افزار روی‌برد ماهواره
- قابلیت استفاده مجدد این نرم‌افزار در پروژه‌های دیگر و نیز توسعه بیش‌تر آن مطابق با نیاز خاص آن پروژه
- پیاده‌سازی نرم‌افزار لایه پایین (راه‌اندازهای سخت‌افزار) بر اساس الزامات سیستم‌های بلادرنگ و با قابلیت اطمینان بالا با توجه به امکانات سیستم عامل بلادرنگ

1 MOP(Mission Operation Plan)
2 FDIR
3 BootLoader

۳- درس آموخته‌ها

چالش ۱

یکی از چالش‌هایی که در حین کار در پروژه ماهواره پارس با آن مواجه شدیم، نداشتن اطلاعات کافی از سلامت باتری بخش توان الکتریکی ماهواره بود که به صورت ناگهانی باعث ایجاد مشکلی در بخش توان ماهواره (به صورت مشخص باتری) شده بود. این اتفاق تجربه‌ای به تیم پروژه داد تا در طول فرایند تست ماهواره، همواره باید از سلامت باتری اطمینان کافی داشت زیرا فازهای مختلف فرایند توسعه و تست ماهواره، دارای تست‌های خاص زمان‌بر و طولانی است.

راهکار ارائه شده چالش ۱

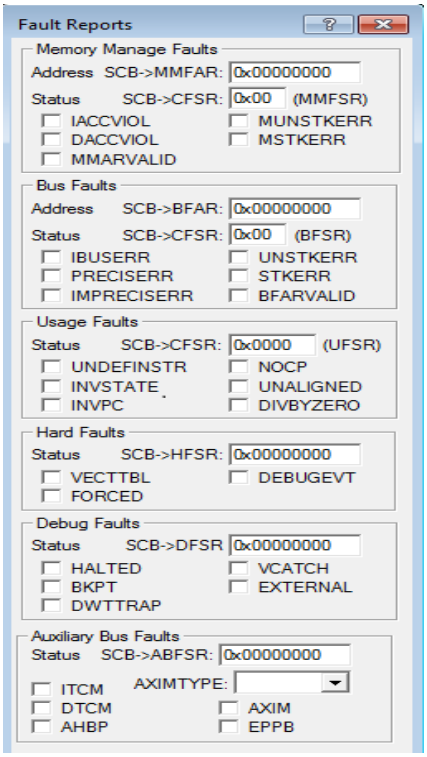
لازم است که حتماً یک روال پایش سلامت ماهواره حین تست‌های نرم‌افزار تهیه شود تا تست‌ها مطابق با آن انجام شود.

درس آموخته ۱

از آنجاکه نرم‌افزار رویبرد ماهواره در طول فرایند توسعه و تست، در فازهای مختلفی از فعالیت‌های تست، دارای تست‌های خاص زمان‌بر و طولانی است، از این‌رو، نیاز به یک رویه پایش سلامت ماهواره است تا بتوان طبق آن از سلامت المان‌ها و ادوات ماهواره اطمینان حاصل کرد. به طور مثال، باید پایش متوالی روی پارامترهای ولتاژ و جریان باتری داشته باشیم تا از سلامت باتری اطمینان کافی حاصل شود. به ویژه زمانی که به طول عمر خود نزدیک شده باشد.

چالش ۲

در فازهای نهایی توسعه ماهواره پارس یک در تست‌های طولانی مدت نرم‌افزار، پیغام استثناء نامشخص[1] مشاهده شد که در زمان‌های طولانی و غیرثابتی اتفاق می‌افتاد (به‌صورت تصادفی). با قرار دادن نقطه شکست[2] در تابع مدیریت استثناء و فعال کردن گزینه "گزارش خطاها" در نرم‌افزار کیل[3] با اعلام خطای غیرصریح[4]) مواجه بودیم. اکثر بخش‌های کد از نظر رفع هشدارهای[5] کامپایلر، دسترسی به حافظه‌ها مانند دستورهای کپی و تنظیم حافظه و استک وظیفه‌های نرم‌افزاری مورد بررسی قرار گرفت اما به علت، غیرصریح بودن نوع خطا، حجم بالای نرم‌افزار، استفاده از حد بالای بهینه‌سازی کامپایلر و پیشامد تصادفی و در زمان‌های طولانی خطا، بررسی مشکل نتیجه‌ای دربر نداشت.



شکل ۲. نمایی از پنجره گزارش خطا در محیط برنامه‌نویسی Keil

راهکار ارائه شده چالش ۲

راه‌حلی که به عنوان روش غیرمستقیم برای رفع حل مشکل ارائه شد، به‌روزرسانی نسخه کامپایلر بود. با انجام این کار با هشدارها و خطاهای جدید در زمان کامپایل کد مواجه شدیم، از جمله خطاهایی در مورد دستورهای تنظیم[6] ساختارهای داده! رفع خطاها و هشدارهای جدید مشکل استثناء را حل کرد.

1 Exception
2 Breakpoint
3 Keil
4 IMPRECIS ERR

5 Warning

6 Alignment

درس آموخته ۲

به نظر می‌رسد فناوری‌های ارائه شده (سخت‌افزار و نرم‌افزار) خالی از ایراد نیستند. به خصوص در پروژه‌های بزرگی مانند توسعه کامپیوتر روی‌برد که از اکثر منابع سخت‌افزار و نرم‌افزار به‌صورت هم‌زمان استفاده می‌شود، احتمال دیده شدن چنین خطاهای بی‌دلیلی وجود دارد. البته معمولاً ضعف طراحی سخت‌افزار و نرم-افزار توسط مجری دلیل اصلی چنین مواردی است ولی با توجه به تجربیات قبلی موضوع اشکال[1] های کامپایلر و سخت‌افزار منتفی نیست، به ویژه هنگام استفاده از قابلیت‌های بهینه‌سازی کامپایلر. بنابراین، بهتر است حین توسعه از نسخه به‌روز کامپایلر استفاده شود. هم‌چنین سند "سیلیکون ایراتا[2]" مربوط به نسخه سخت‌افزار مورد توجه قرار گیرد.

چالش ۳

در تست‌های متوالی ماهواره پارس یک، این موضوع دیده شد که برخی المان‌ها یا ادوات ماهواره روی یکدیگر تأثیرگذار هستند یا باعث ایجاد تداخل در سیستم می‌شوند. به طور مثال، زمانی که قرار بود تست سناریوی تصویربرداری هم زمان دوربین‌های ماهواره انجام شود، تأثیر نادرست دوربین‌ها روی همدیگر دیده شد. بدین صورت که با خاموش شدن اولین دوربین و اتمام تصویربرداری، باند ایکس نیز خاموش می‌شد و این درست نبود! مثال دیگری از تأثیرگذاری نادرست دوربین‌ها روی همدیگر این بود که دوربین اصلی و رزرو نباید با هم روشن شوند و کار کنند!

راهکار ارائه شده چالش ۳

راهکار ارائه شده برای مشکلات مربوط به تأثیرگذاری دوربین‌ها به این ترتیب بود که در کد نرم‌افزار، شرطی برای مدیریت چهار مود تصویربرداری که شامل مودهای بلادرنگ/ شبه بلادرنگ/ ارسال که نیاز به فعال بودن باند ایکس دارند و مود ذخیره که نیاز به باند ایکس ندارد و فقط خود دوربین درگیر است؛ پیاده‌سازی شد.

درس آموخته ۳

در طول فرایند تست‌های ماهواره پارس یک، برخی المان‌ها یا ادوات ماهواره روی یکدیگر تأثیرگذار هستند یا اینکه روی هم تداخل ایجاد خواهند کرد که باعث ایجاد اشکال در سیستم می‌شوند. به طور مثال، تأثیر متقابل مگنتو تورکرها و مگنتو میترها روی همدیگر یا تأثیر دوربین‌ها روی همدیگر (که در تست سناریو تصویربرداری هم‌زمان دوربین‌ها مشاهده شد) نمونه‌هایی از این دسته هستند که باعث ایجاد چالش‌هایی در کار شد.

چالش ۴

در طول فرایند تست، یکی از چالش‌های دیگری که با آن مواجه شدیم این بود که گاهی اوقات اینترفیس‌های سخت‌افزاری کامپیوتر روی‌برد عملکرد درستی را انجام نمی‌دادند و به همین دلیل آن المان سخت‌افزاری در وضعیت نامشخص نرم‌افزاری قرار می‌گرفت و با عیب‌یابی[3] کردن نرم‌افزار متوجه این وضعیت نامشخص می‌شدیم و بعد مجبور به ریست کردن نرم‌افزار روی‌برد و انجام مجدد تست می‌شدیم.

راهکار ارائه شده چالش ۴

برای حل چالش به وجود آمده، تصمیم گرفته شد تا برای اینترفیس‌های سخت‌افزاری از جمله CAN، الگوریتم‌های تشخیص خطا[4] پیاده‌سازی شود. این الگوریتم‌های تشخیص خطا به راحتی خطای ایجاد شده را شناسایی کرده و با رفع مشکل و دریافت پاسخ مورد انتظار از اینترفیس سخت‌افزاری CAN، روال تست ادامه می‌یابد. الگوریتم تشخیص خطا برای اینترفیس‌های سخت‌افزاری CAN بدین ترتیب است که پرچمی[5] به نام 'عدم وجود داده' تعریف می‌شود، در صورت فعال شدن این پرچم یا ۱ شدن آن، (یعنی آن المان داده‌ای از سخت‌افزار دریافت نکرده تا پاسخی بدهد!) و عملیات ریست شدن آن المان انجام می‌شود،

1 Bug
2 silicon errata
3 Debug

4 Fault Detection
5 Flag

یعنی المان پاسخ مناسبی از سخت‌افزار دریافت نکرده بنابراین، ریست می‌شود و مجدداً دستور به سخت‌افزار مورد نظر ارسال می‌شود تا روال تست به درستی و کامل انجام شود.

درس آموخته ۴

یکی از مهم‌ترین مواردی که باید در طول تست‌ها مورد توجه قرار داد، در نظر گرفتن مشکلات غیرقابل پیش‌بینی در الگوریتم‌های تحمل‌پذیری خطا[1] است. به طور معمول در الگوریتم‌های تشخیص خرابی[2] زیرسیستمی، داده‌های سلامت زیرسیستم مانند ولتاژ، جریان، دما و ... بررسی می‌شوند و الگوریتم تشخیص خرابی یا خطا و عیب‌یابی آن توسط زیرسیستم مربوطه ارائه می‌شود.

اما یکی از تجربیات مهمی که در حین تست برای تیم نرم‌افزار و سخت‌افزار کسب شد این بود که علاوه بر بررسی داده‌های سلامت و حیاتی هر زیرسیستم در الگوریتم‌های تحمل‌پذیری خطای آن، مشاهده عدم دریافت پاسخ مورد انتظار از المان‌های سخت‌افزاری برای اجرای الگوریتم‌های تشخیص خرابی باعث شد که یک بخش چک کردن عدم وجود داده[3] در ابتدای هر الگوریتم تشخیص خرابی اضافه کنیم به این صورت که با عدم گرفتن پاسخ در طول یک بازه زمانی مشخصی مثلا یک دقیقه (در این بازه متناسب با نرخ نمونه‌برداری[4] هر المان سخت‌افزاری متوجه تعداد دفعات عدم پاسخ‌گیری می‌شدیم که برای هر زیرسیستم متفاوت بود!). توضیح اینکه نرخ نمونه همان نرخ نمونه‌برداری داده از سخت‌افزار است و نرخ HK[5]، نرخ نمونه‌برداری داده گزارش شده در قالب تله‌متری به ایستگاه زمینی است (منظور تله‌متری‌های مشاهده شده در زمین است که با درخواست تله‌کامند از زمین با نرخ مشخص و ثابت HK Rate به زمین ارسال می‌شوند). اگر داده‌ای از سخت‌افزار خوانده نمی‌شد، آن المان را ریست می‌کردیم (به طور مثال با سه بار ریست شدن المان، پرچم 'عدم وجود داده' چک می‌شد) و به محض گرفتن پاسخ از سخت‌افزار، روال تست ادامه می‌یابد.

چالش ۵

در حین انجام تست‌های روی میز[6] ماهواره، لازم بود طبق برنامه گانت، تست‌های مربوط به پردازش/ نرم‌افزار در حلقه[7] نیز انجام شود. از آنجاکه تست‌های روی میز ماهواره در بازه زمانی حساسی بود و باید به ناظرین تحویل داده می‌شد و امکان متوقف کردن کار وجود نداشت، تصمیم گرفته شد یک بستر جداگانه برای تست‌های پردازش/ نرم‌افزار در حلقه در نظر گرفته شود. این کار با اعمال تغییرات جزیی روی برد ارزیابی[8] صورت گرفت و یک بستر جداگانه برای این تست در نظر گرفته شد.

راهکار ارائه شده چالش ۵

با اعمال تغییرات جزیی روی برد ارزیابی، یک بستر کامپیوتر رویبرد، نرم‌افزار رویبرد و ایستگاه زمینی ایجاد شد تا بتوان تست‌های پردازش/ نرم‌افزار در حلقه را جداگانه انجام داد.

درس آموخته ۵

به دلیل سرعت بخشیدن به تست‌های ماهواره پارس یک، تصمیم گرفته شد، یک بستر جداگانه برای تست پردازش/ نرم‌افزار در حلقه و یک بستر دیگر برای تست‌های روی میز ماهواره ایجاد شود تا به صورت موازی هر یک از این تست‌ها به طور جداگانه انجام شود. در واقع، هدف "ایجاد یک بستر موازی برای تست‌های هم‌زمان و موازی بستر پردازش/ نرم‌افزار در حلقه و بستر روی میز ماهواره" است. با ایجاد تغییرات جزیی در نرم‌افزار روی برد ماهواره و بارگذاری آن روی برد ارزیابی، بستر جداگانه‌ای برای تست‌های پردازش/ نرم‌افزار در حلقه ایجاد شد. این کار در مدیریت بهینه زمان بسیار تأثیرگذار بود.

1 FDIR
2 FD
3 No Data
4 Sample Rate
5 HK Rate
6 Flat
7 Processor In-the- Loop(PIL)
8 Evaluation Board

۵- دستاوردها

دستاورد ۱

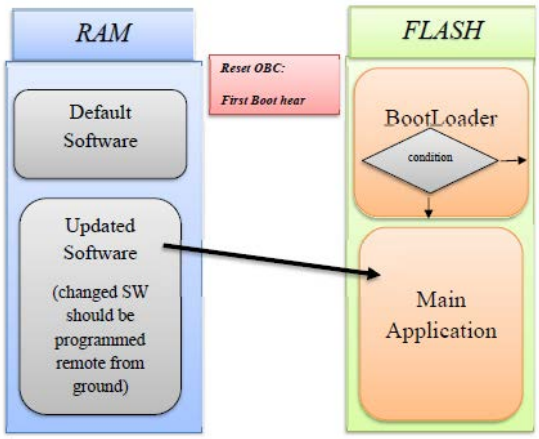
با توسعه نهایی نرم‌افزار روی‌برد ماهواره پارس یک مطابق با الزاماتی که از طرف تیم ناظر ارائه شد، این امکان و قابلیت در ماهواره پارس یک وجود دارد که بتوان برای یک ساختار داده[1] مشخص در یک زمان مشخص (تاریخ و ساعت مشخص) به مدت سه روز متوالی، داده‌های HK آن ساختار داده در فایلی ذخیره شود. به عبارتی امکان ذخیره‌سازی تا سه روز برای داده‌های HK ماهواره پارس یک وجود دارد. از آنجاکه چنین قابلیت و امکانی در ماهواره‌های توسعه یافته در پژوهشکده تا به اینجا دیده نشده است، می‌توان به این قابلیت به عنوان یک دستاورد مهم و کلیدی در ماهواره پارس یک اشاره کرد.

دستاورد ۲

یکی دیگر از قابلیت‌های توسعه داده شده در نرم‌افزار روی برد ماهواره پارس یک، آپلود نسخه بهبود یافته و بهینه شده همراه با افزودن قابلیت‌های جدید با عنوان به‌روزرسانی شده از طرف کاربر زمینی از سمت زمین، به صورت ریموت در ماهواره قرار گرفته در مدار است. بدین ترتیب می‌توان یک فایل باینری که شامل نسخه جدیدی[2] از نرم‌افزار روی برد ماهواره است را از ایستگاه زمینی به ماهواره ارسال کرد (آپلود نرم افزار).

نکته مهم‌تر اینکه این امکان بر مبنای سرویس شماره ۱۳ (Large Data Service) پروتکل PUS[3] در استاندارد ECSS پیاده‌سازی و تست شده است (البته ناگفته نماند که به طور کلی نرم‌افزار ماهواره پارس یک بر مبنای سرویس‌های همین پروتکل PUS در استاندارد ECSS توسعه یافته و این موضوع هم یکی از مشخصات بارز و کلیدی این ماهواره است).

در واقع هدف از این پیاده‌سازی این سرویس، به نحوی مدیریت حافظه است، بدین ترتیب که نسخه جدید نرم‌افزار که قرار است در حافظه فلش ماهواره بنشیند، ابتدا در حافظه MRAM (در آدرس مخصوص) نوشته شده و سپس به حافظه فلش منتقل می‌شود (در فلش کپی می‌شود). بعد از اتمام انتقال نرم‌افزار در فلش، بعد از یک بار ریست شدن نرم‌افزار فلش، از راه‌انداز نرم‌افزار یا بوت لودر بالا می‌آید و نرم‌افزار بخش اصلی[4] فلش اجرا می‌شود که شامل همان نرم‌افزار به‌روز شده است. این پروسه از سمت ایستگاه زمینی با روال زیر توسط کاربر زمینی انجام می‌شود: ذخیره نرم‌افزار به روز شده در فضای آدرس مشخصی از حافظه MRAM با قابلیت اطمینان بالا، انتخاب نرم‌افزار اصلی اولیه[5] یا به‌روز شده توسط کاربر زمینی به منظور اجرای آن در فلش، اجرای نرم‌افزار راه‌انداز یا بوت لودر بعد از ریست کامپیوتر روی برد (OBC)[6] ، انتقال نسخه نرم‌افزار به‌روز شده از حافظه MRAM به حافظه فلش[7]. از آنجاکه چنین قابلیت و امکانی در ماهواره‌های توسعه یافته در پژوهشکده تا به اینجا دیده نشده است، می‌توان به این قابلیت نیز به عنوان یک دستاورد مهم و کلیدی در ماهواره پارس یک اشاره کرد.



شکل ۳. نحوه ذخیره‌سازی و اجرای نرم‌افزار روی حافظه فلش و RAM کامپیوتر روی‌برد ماهواره پارس

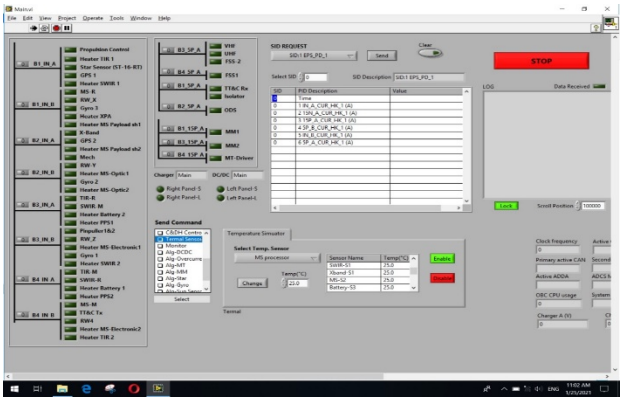
دستاورد ۳

یکی دیگر از قابلیت‌های توسعه یافته در نرم‌افزار روی برد ماهواره پارس یک، تهیه یک بستر تست به منظور تزریق خرابی یا

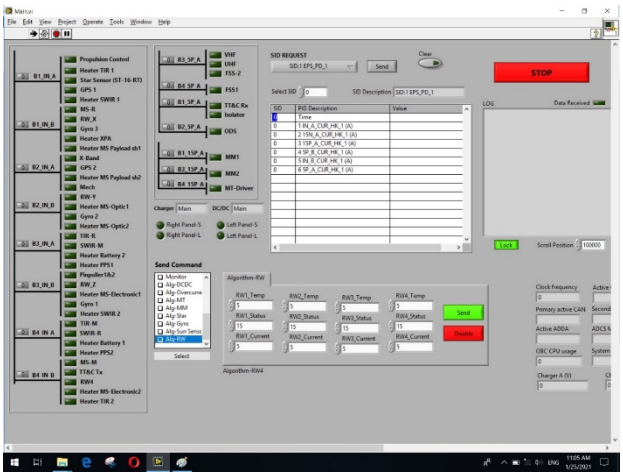
1 Structure Identification
2 Updated Version
3 Packet Utilization Standard
4 Main

5 Default
6On Board Computer
7 FLASH

خطا برای تست الگوریتم‌های تشخیص خرابی زیرسیستم‌ها است که با تعامل بین توسعه دهنده نرم‌افزار روی برد و توسعه دهنده نرم‌افزار تستر ماهواره انجام شد (به طور مثال برای چک کردن الگوریتم تشخیص خرابی حرارت، پنل تزریق خرابی را در نرم‌افزار تستر ایجاد کردیم و از طریق آن الگوریتم تشخیص خرابی حرارت را صحه‌گذاری کردیم). توسعه این قابلیت کمک بسیار زیادی به تیم نرم‌افزار و سیستم، در طول پروسه کامل تست نرم‌افزار روی برد ماهواره پارس یک کرد. نقش کلیدی این توسعه باعث شد که به عنوان یک دستاورد در پروژه دیده شود.



شکل ۴. نمایی از نرم‌افزار تستر ماهواره پارس یک



شکل ۵. نمایی دیگر از نرم‌افزار تستر ماهواره پارس یک

۵- نتیجه‌گیری

نرم‌افزار روی‌برد ماهواره به تنهایی گستردگی و پیچیدگی‌های خاص خود را دارد که مستلزم تلاش و پشتکار و دقت زیاد افراد تیم است. در طول فرایند طراحی، پیاده‌سازی و تست نرم‌افزار روی‌برد ماهواره پارس یک، مباحث و نکات زیادی استخراج شد که لازم است سند مفصلی در این مورد تهیه شود. با این وجود، مقاله حاضر سعی بر آن داشت تا به ارائه خلاصه‌ای از درس آموخته‌ها و دستاوردهای حین تست که نسبت به بقیه موارد مهم‌تر و چالشی‌تر بوده‌اند، بپردازد. به یقین، استفاده از این تجربیات و درس آموخته‌ها کمک بسیار زیادی در هموار کردن مسیر برای پروژه‌های آینده خواهد کرد.

مراجع

[1] J. Eickoff, OnBoard Computers, OnBoard Software and Satellite Operations, ‘Springer Aerospace Technology’ 2012.

[2] J. Eickhoff, The FLP Microsatellite Platform, Flight Operations Manual, ‘Springer Aerospace Technology 2016’

[3] A Comparative survey on flight software frameworks for ‘new space nanosatellite missions’, J. Aerosp. Technol.Manag.,Sao Jose campos, v11,e4619,2019.

[4] essr.esa.int/project/osra-onboard-software-reference-architecture,updated on 2021.

[5] New Concepts for onboard software development, 2022.

[6] Pars1 OBSW Source Code- Final Version 2019.