נושא :סיכום פעילות מקצועית במסגרת עבודתי בתע"א

שלום רב,

ברצוני להציג בפניכם סיכום מקיף של תרומתי המקצועית במסגרת תפקידי כמהנדס תוכנה בתעשייה האווירית.

מאז שהצטרפתי לתע"א בדצמבר 2015, פעלתי בעקביות לפיתוח והטמעה של פתרונות תוכנה מתקדמים, הן במערכות זמן-אמת והן באפליקציות מחקריות ודסקטופ, תוך הובלה של מחזורי פיתוח מלאים – מאפיון הדרישות, דרך עיצוב הארכיטקטורה, מימוש, בדיקות, אינטגרציה ועד תיעוד טכני מקיף.

מחלקת תוכנה מוטסת (דצמבר 2015 – אוקטובר 2020)

בתקופה זו פיתחתי לוגיקת שליטה חדשה משלב אפיון ועד מימוש בקוד בשפת תכנות C תוך שיתוף פעולה הדוק עם מהנדסי אוויוניקה על ניסוח דרישות ותיעוד טכני. הלוגיקה עסקה בעיקר בניהול מנוע, דלק ומשאבת דלק.

יוזמתית, נכנסתי לעובי הקורה ושדרגתי מקטעים משמעותיים בתוכנת השליטה של המל"ט, שנמצאה במצב קשה מבחינה תחזוקתית. בין פעולותיי הבולטות:

- בנייה מחדש וסטנדרטיזציה של מודולים קריטיים: כניסות/יציאות דיגיטליות, קווים סריאליים, זיכרון לא נדיף, תקשורת בין מחשבים AVCs, ניהול אנטנה.
- שיפור ביצועים עבור מעבדי PowerPC ושדרוג סביבת הפיתוח לגרסאות מתקדמות של GHS Multi.
- ארגון מחדש של עץ הספריות, סידור קבצים לפי נושאים התקנים.
- הסרת קוד ישן, לא פעיל ומיותר, יישור סגנון קוד, סטנדרטיזציה של שמות קבצים ומשתנים, שיפור קריאות, קוהרנטיות וקופלינג נמוך.
- פיתוח ואינטגרציה של יכולות חדשות לשיפור אמינות ובטיחות תפקוד המערכת.
- פיתוח אפליקציית ניטור אוויוני (Avim) חדשה בשפת תכנות ++C עם Qt.

מחלקת אלגוריתמים (נובמבר 2020 – דצמבר 2021)

פיתוח מודול ניהול מרכזי בפרויקט נחיתת תמונה מבוסס על מערכת הפעלה ווינדוס בשפת תכנות ++C כולל איסוף דרישות, עיצוב, מימוש, בדיקות ותיעוד.

מחלקת רובוטיקה – אלתא (ינואר 2022 – יוני 2022)

פיתוח מודול ניהולי-מרכזי בפרויקט השקפת עולם מבוסס על מערכת הפעלה וינדוס בשפת תכנות ++C כולל ניהול תקשורת פנימית של תת-מערכות רובוטיות.

מחלקת תוכנת מוצרים (יולי 2022 – יולי 2023)

ביצוע מחקר טכנולוגי ופיתוח יישומים ניסיוניים במספר שפות תכנות C, C++, C#, Python.

מחלקת תוכנת מוצרים (יולי 2023 – ינואר 2024)

פיתוח יישום ייצור זמן אמת בפרויקט GVDM מבוסס על מערכת הפעלה ווינדוס בשפת תכנות ++C. התוכנה נמצאת כיום בשימוש חיל האוויר.

מחלקת תוכנת מוצרים (פברואר 2024 – היום)

פיתוח יישום עיבוד אותות למערכת משובצת של חברת UEI מבוססת על מערכת הפעלה לינוקס בשפת תכנות ++C.

חונכות ושיתופי פעולה

במהלך כל שנותיי בתע"א, תרמתי גם בהדרכה וליווי של מהנדסי תוכנה צעירים, שיתפתי פעולה עם מהנדסי אוויוניקה במגוון תחומים טכניים, ותמכתי בהבנה של קוד ומסמכים מורכבים.

מכתבי המלצה והערכות

קיימים ברשותי מכתבי המלצה ודוחות הערכה רבים שקיבלתי לאורך השנים מלקוחות, קולגות, ומנהלים בתע"א – המלצות המשקפות את המחויבות, המקצועיות, וההשפעה הרחבה של עבודתי.

בברכה,

מיכאל ז'אק
מהנדס תוכנה, תע"א
mjack@iai.co.il | 052-507-7979

To: Human Resources Department
From: Mikhail Jacques
Subject: Summary of Professional Contributions and Activities

Dear HR Team,

I am writing to provide a comprehensive summary of my professional contributions and responsibilities during my tenure at Israel Aerospace Industries (IAI), in support of my candidacy for promotion.

Since joining IAI in December 2015 as a Software Engineer, I have consistently delivered high-impact solutions across multiple departments, demonstrating both technical depth and leadership. I have led and participated in full software development life cycles—from initial requirements and architectural design to implementation, integration, and documentation—delivering reliable, real-time, and research-grade applications.

UAV Real-Time Software Department (Dec 2015 – Oct 2020)

During this period, I spearheaded the development of new non-avionic control logic from scratch in C, collaborating closely with avionics engineers to co-author specifications and technical documentation. My work primarily focused on governing critical UAV subsystems such as engine, fuel, and fuel pump operations.

I also took the initiative to systematically refactor and elevate large portions of the UAV plane-control codebase, which had suffered from technical debt and architectural drift. Among the key contributions:

- Rebuilt and standardized entire modules, including Digital I/O, Serial Lines, Non-Volatile Memory, and AVC-to-AVC communication.
- Consolidated scattered logic and files into a clean, hierarchical structure that now adheres to high-cohesion, low-coupling principles.
- Introduced code efficiency improvements specifically optimized for PowerPC CPUs.
- Re-architected the watchdog logic and added robust exception logging mechanisms.
- Improved code readability and maintainability through naming conventions, modularization, and rigorous documentation.
- Removed obsolete, redundant, and dead code across the codebase, significantly enhancing maintainability.
- Upgraded development environment compatibility to GHS Multi 6.1.6 and 7.1.4.
- Developed a new Avionics Monitoring (Avim) application in C++ using the Qt framework.

Algorithmics Department (Nov 2020 – Dec 2021)

Here, I designed and implemented from the ground up a central management module in C++ for a Windows-based Image Landing research project. The work included requirements analysis, software architecture, implementation, testing, and documentation.

Robotics Department – Elta Systems (Jan 2022 – July 2022)

I designed and implemented another C++ Windows-based central module for the World Perception project. My work integrated and managed communication among various robotic subsystems in a research-driven environment.

Integration Department (July 2022 – July 2023)

I explored cutting-edge technologies and delivered several experimental applications in C, C++, C#, and Python, supporting diverse research initiatives.

Engineering Department (July 2023 – Jan 2024)

I designed, developed, and deployed a real-time, Windows-based Ground Video-Data Multiplexer (GVDM) message routing system, now actively used by the Israeli Air Force. This high-performance system was built from scratch in C++ and involved full-cycle development from requirements to deployment.

Current Role – Integration Department (Jan 2024 – Present)

I am currently developing a Linux-based C++ signal processing application, continuing to contribute to mission-critical research and development initiatives.

Mentorship and Collaboration

Throughout my tenure, I have consistently collaborated with avionics engineers, provided technical mentorship to junior developers, and facilitated cross-team knowledge transfer. My contributions extend beyond coding to fostering clarity in documentation, code comprehension, and architectural reasoning.

I am proud of my continuous commitment to engineering excellence, innovation, and mentorship at IAI. I believe that my diverse technical achievements, initiative, and leadership qualify me for a role of greater responsibility, and I respectfully submit this record in consideration for promotion.

Thank you for your time and attention.

Sincerely,
Mikhail Jacques
Software Engineer, IAI
mjack@iai.co.il | 052-507-7979

I have formally worked as software engineer since July 2012. I have been employed as a software engineer at IAI since December 2015.

From December 2015 until October 2020, I worked as a software engineer in the UAV plane-control real-time software department. During this time, I developed and implemented a new non-avionic programming logic from scratch in the C programming language. I co-authored requirements and related technical documentation with avionics engineers. The non-avionic programming logic dealt, for the most part, with governing and monitoring the engine, fuel, and fuel pump functionalities. On top of that, I took the initiative to improve what was in my power in the UAV plane-control real-time software, which was a real mess, to say the least. The following is a list of things that I am responsible for fixing and upgrading. The list is composed of highlights and is not exhaustive

- Rebuilt, standardized the entire Avim module,
- Rebuilt, standardized and documented the entire digital inputs/outputs module,
- Rebuilt, standardized and documented the entire serial lines module,
- Rebuilt and standardized the entire AVC-to-AVC communication module,
- Rebuilt, standardized and documented the entire Non-Volatile memory (NvRAM) module,
- Developed, integrated and tested new programming logic that now saves both the first and the last exception information in NvRAM,
- Improved and standardized Watchdog programming logic,
- Added antenna (it has neither been tested nor formally documented yet),
- Rebuilt and standardized the entire directional antenna module (it has neither been tested nor formally documented yet),
- Standardized tasks and connections,
- Standardized code writing signature where applicable and possible,
- Standardized the code as much as I am allowed by ICD and flight control constraints,
- Consulted with Elta Systems Ltd. documentation and with local hardware engineers regarding proper terminology and then re-implemented it to adhere to commonly accepted standards,
- Removed obsolete OPT code,
- Removed tons of dead code,
- Removed unused files,
- Removed duplicate files,
- Removed redundant tasks,
- Removed duplicate definitions within the files,
- Re-aligned code to look the same throughout the entire project,
- Renamed and standardized files to reflect the essence of their purpose,
- Renamed and standardized variable names,
- Optimized references to files and consolidated them in one place,
- Consolidated data in files with strict adherence to key software engineering principle of high coherence and low coupling,

- Improved code efficiency from the PowerPC CPU perspective,
- Added new features and improved operational program safety,
- Upgraded code to develop and compile using GHS Multi 6.1.6. and Multi 7.1.4 as opposed to old Multi 5.0.6,
- Reconstructed the entire project's hierarchical tree structure because it was obsolete, messy and inefficient,
- Relocated all address spaces into one folder, namely APPLS
- Relocated all the software modules that deal with devices (LRUs) from wherever they were scattered into the Devices folder under the Main address space,
- Relocated all the ICD generated files from wherever they were scattered into the dedicated ICD directory,
- Discovered numerous code defects along the way and fixed them,
- Documented code via self-descriptive names of files, functions and variables,
- Developed and partially documented a code standard that anyone working on MK2 software or its derivatives should adhere to.

In addition, in the fall of 2017, I designed and developed a brand-new Avionics monitoring (Avim) application from scratch in C++ using the Qt framework.

From November 2020 until December 2021, I worked in the Algorithmics department. During that time, I gathered and analyzed the requirements, designed and developed from scratch in C++, tested and documented a central management module for the Windows-based Image Landing research project.

From January 2022 until July 2022, I worked in the Robotics department of Elta Systems. During that time, I analyzed the requirements, designed and developed from scratch in C++, tested and documented a Windows-based central management and communication module for the World Perception research project.

From July 2022 until July 2023, I mostly researched new technologies and implemented a few small research applications for the Integration department in a wide variety of technologies and programming languages, such as C, C++, C#, and Python.

From July 2023 until January 2024, I gathered and analyzed the requirements, prepared requirements documentation, designed architecture, developed from scratch in C++, tested, integrated, and documented a Ground Video-Data Multiplexer (GVDM) message routing real-time Windows-based production application for the Engineering Department that is currently used by the Israeli Air Force.

During all these years, I also assisted various avionics engineers with documentation and code understanding and assisted and tutored less experienced software engineers in various technologies and coding assignments.

Right now, I am developing a Linux-based signal processing application in C++ for the Integration Department.