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 in C programming language a brand new non-avionic programming logic from scratch as well as co-authored with avionics engineers’ requirements and related technical documentation. The non-avionic programming logic dealt for the most part with governing and monitoring of engine, fuel and fuel pumps functionalities. On top of it, I took the initiative to improve what has been in my power in the UAV plane-control real-time software which was a real mess, to say the least. 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 in order 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 sort of code standard that anyone who works on MK2 software or its derivatives should adhere to.

In addition, in the fall of 2017 I designed and developed from scratch a brand-new Avionics monitoring (Avim) application in C++ programming language using 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 diverse variety of technologies and programming languages, e.g. 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 application for the Engineering Department.

During all these years I also assisted various avionics engineers with documentation and code-understanding as well as 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.