MERT EFE ŞENSOY

COMPUTER ENGINEERING STUDENT

CONTACT

Phone: +90 505 422 01 11
Email: mefe.sensoy@tedu.edu.tr
GitHub: https://github.com/mertefesensoy
LinkedIn: www.linkedin.com/in/mert-efe-sensoy
Website: https://www.mertefesensoy.dev/
Address: Ankara, Turkey

EXPERIENCES

Turkish Airlines - Junior Data Analyst (Part Time) **September 2025 - September 2026**

Development of Python-based data infrastructure for the Revenue Analysis and Reporting Unit. Backend for Dynamic Discrete Surcharge Prediction (DDSP): an Airflow-orchestrated ELT process from Teradata and Oracle into a PostgreSQL medallion architecture; design of leakage-free feature views computed as of the decision time (t) for a CatBoost model that maximises RASK at the lowest possible surcharge and therefore recommends a capacity-conditional optimal surcharge, together with a FastAPI service exposing them. An auditable and reproducible decision-support pipeline replacing the ad-hoc SPSS Modeler flow.

ASELSAN - Summer Internship **July 2026 - August 2026**

Development of a C++17 code-generation and mission-execution system for tactical messaging, within the Signal Processing and Embedded Software System Design Directorate. Compilation of a spreadsheet-defined data model into compilable C++ headers via a canonical intermediate representation, with a hard validation boundary and an atomic write contract guaranteeing that not a single byte of output is produced unless the model validates in full. Publish-subscribe topology over an AMQP broker, a dead-letter queue, and resilience testing under broker outages. Scheduling of conflicting orders through a fixed-size worker pool, a per-resource lease model, priority classes, and step-boundary preemption.

ASTECH Defence - Summer Internship **July 2025 - September 2025**

Development of a FreeRTOS-based real-time embedded application in embedded C, managing more than 10 concurrent tasks through priority-based scheduling, queues, and mutexes. Runtime and memory analysis (stack and heap monitoring) to verify deterministic and robust system behaviour under 90% CPU load; application of defence-industry software quality standards, static analysis methods, and Git/GitHub-based version control processes.

PROJECTS

SuperconducTED - Noise Engine for Quantum Emulation - Project lead **March 2026 - Ongoing**

Lead a five-person team building an open-source Takagi-Sugeno-Kang fuzzy inference layer over Qiskit Aer that represents the epistemic uncertainty Aer's crisp NoiseModel parameters (T1, T2, gate and readout error) cannot express, modelling IBM calibration drift as interval-valued noise ensembles. Own the architecture of the six-stage pipeline (calibration ingestion, feature extraction, fuzzification and TSK rule firing, defuzzification, squashing, channel projection) and of an Aer Factory/Ensemble integration that realises uncertainty at ensemble construction time, working around Aer's lack of per-shot Python hooks. Maintain the calibration-snapshot polling CLI accumulating historical records required for ANFIS training. The end-to-end pipeline and its architectural validation are complete; membership-function training and non-degenerate ensemble execution are still in progress. MIT-licensed, CI-tested on Python 3.11/3.12, with an ADR ledger.
github.com/SuperconducTED/superconducted-noise-engine

zbridge-asm-lab **February 2026 - Ongoing**

Built the Phase 1 groundwork for calling MVS services from Go assembly without cgo, targeting the interface WTO(message string) error via SVC 35 on z/OS; mentored by Jürgen Holtz (IBM Senior Technical Staff Member) of IBM Böblingen. Completed six exercises: Plan 9 assembly toolchain validation, EBCDIC IBM-1047 byte translation, length-prefixed parameter list construction, register and stack frame layout, oracle-tested byte comparison, and a Linux SYSCALL trap as the rehearsal for SVC 35. Documented each with an s390x stub covering the corresponding target instruction (TR, STH/MVC, CLC) and the MVS R13/R14/R15 linkage conventions. Verified on amd64 with go vet and go test; nothing has yet run on z/OS. Set up a two-track Hercules laboratory (MVS 3.8j and QEMU s390x) to test parameter list correctness ahead of mainframe access.
github.com/mertefesensoy/zbridge-asm-lab

The Open Memory Initiative (OMI) **December 2025 - June 2026**

Founded OMI, an open engineering effort producing a reviewable DDR4 UDIMM reference design (8 GB, 1R, x8, non-ECC). Authored the charter, scope, and staged methodology, and completed schematic capture across power/PDN, address/command/clock (fly-by), DQ/DQS, SPD, and the 288-pin edge map. Also published spdr, a no_std Rust DDR5 SPD decoder and semantic linter that validates module timing and structure beyond CRC. Licensed under CERN-OHL-S-2.0, CC BY-SA 4.0, and Apache-2.0.
github.com/The-Open-Memory-Initiative-OMI **crates.io/crates/spdr**

ACADEMIC BACKGROUND

TED University **2023 - 2027**

Computer Engineering (English / 100% Scholarship)

- Business Administration Minor Program (English)
- IBM Z Student Ambassador & IBM Z Student Club President
- Student Mentor, Center for Teaching and Learning (CTL) Mentor Support Program **September 2024 - Ongoing**
- Lab Tutor, CMPE 113 (Introduction to Programming in Java) **Spring 2025**

AWARDS AND CERTIFICATES

- **IBM Champion ’26** - A selective annual recognition for technical leaders contributing to the IBM Z and broader IBM community through content, mentorship, and advocacy. **IBM**
- IBM Data Science Professional Certificate **IBM**
- IBM Z Xplore - Advanced **IBM**
- Network Automation Engineering Fundamentals **Cisco Learning and Certifications**
- Unmanned Aerial Vehicle (UAV) Pilot Certificate **SHGM / Directorate General of Civil Aviation**

SKILLS

- Programming Languages: Python, Java, C, C++, Go, Rust, Assembly (s390x/amd64), SQL
- ML / Data: CatBoost, scikit-learn, pandas, NumPy, DuckDB, Great Expectations (GX), Matplotlib
- Backend / Infra: FastAPI, Node.js, SQLAlchemy, PostgreSQL, Docker
- Embedded / Hardware: FreeRTOS
- Tools: Git, GitHub Actions, Linux
- Languages (spoken): English (C1), Turkish (native)