Abrar Rahman Protyasha
abrar.rahman98@gmail.com
+1 917-862-1504
https://aprotyas.github.io

Education

University of Rochester — Rochester, NY
Bachelor of Science in Electrical and Computer Engineering — *Aug. 2017 - May 2021*
- Honors: Tau Beta Pi

University of Rochester — Rochester, NY
Master of Science in Electrical and Computer Engineering — *Aug. 2021 - May 2022*
- Concentration: Robotics

Work Experience

Argo AI — Pittsburgh, PA
Software Engineer, Developer Tools — *June 2022 - Nov. 2022*
- Architected Python libraries to generate documentation graphs enabling requirements traceability across the entire autonomy stack.
- Productionized C++17-based log visualization tool adopted by every autonomy developer, accelerating developer triage cycles by $> 100\%$.
- Designed Bazel tooling for improved clang-tidy integration, lowering memory footprint by $> 500\%$.
- Certified to review Python code, commonly known as "readership" status.

Open Robotics — Mountain View, CA
Software Engineering Intern — *May 2021 - Aug. 2021*
- Developed roadmap features for ROS 2's C++ client API and geometry libraries.
- Redesigned `rosdoc2` - a domain-agnostic tool for API document generation in the ROS ecosystem.
- Reworked code linting infrastructure (in Python) to introduce an extensible file exclusion interface.

Robotics and Artificial Intelligence Laboratory - University of Rochester — Rochester, NY
Research Assistant — *Aug. 2020 - May 2022*
- Developed `cpg-viewer` - a Qt5 program (C++) for the 3D visualization of arbitrary robot locomotion.
- Wrote ROS packages for sensor modules, a wrapper around an actuator device driver, and a high level robot manager node to integrate the Robotis OP2 platform into lab's physical and simulated test beds.

Projects

- **ROS 2**: Open source contributor and reviewer of various packages in the Robot Operating System (ROS) ecosystem. Notable contributions:
  - rosbag2 - New playback method for recorded data.
  - ament_lint - Improved code linting experience - file exclusion, better CMake integration, etc.
  - Trac-IK - Ported the popular inverse kinematics solver, and its Python wrapper, to ROS 2.
- **nodl_to_policy**: Tooling to generate a ROS 2 Access Control Policy from the Node Interface Definition (NoDL) of a ROS system, used in secure robotics applications - such as RoMi-H.
  Technologies: Python3, CLI entry points, XML, XSLT, Security.
- **Autonomous mobile robot software architecture**: Developed ROS packages (C++14) for simulation, perception, occupancy grid mapping, path planning, localization, path following controls, and an OpenGL GUI to explore a partially known world using a TurtleBot2.
  Technologies: C++, ROS, CMake, SLAM, Sampling-based motion planning, Pure pursuit.

Skills

Languages: Proficient in C++ and Python. Conversant in C, Bash, SQL, Starlark.

Tools/Technologies: Git, Linux, Build systems (Bazel, CMake), ROS 2, Visualization frameworks (ImGui, Qt5), Testing frameworks (GoogleTest, Catch2, pytest), CI tools (Jenkins, Github actions).