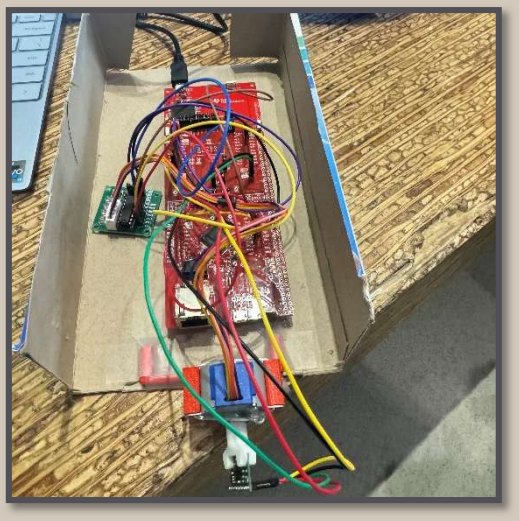
Embedded Spatial Scanner (LiDAR)

"Affordable rotational LiDAR mapping"

BACKGROUND

The MSP432E401Y drives a 28BYJ-48 stepper motor in 11.25° increments via a ULN2003 driver, pausing at each step to capture a millimeter-accurate VL53L1X ToF reading over I2C before wrapping it into 115 200 bps UART frames, with a pushbutton for scan control and LEDs for status. On the host PC, a Python script uses pyserial to convert each polar reading into Cartesian XYZ coordinates, simulating forward motion between slices—and renders a live 3D point cloud in Open3D.



HOW DOES IT WORK?

In our system, the **MSP432E401Y** drives a 28BYJ-48 stepper motor in 11.25° increments via a ULN2003 driver, pausing at each position to capture a millimeter-accurate time-of-flight reading from the VL53L1X sensor over **I2C**. Those measurements are immediately wrapped into **UART frames at 115 200 bps**, with a pushbutton to start and stop scanning and LEDs to indicate system status. On the host PC, a Python script uses pyserial to ingest the serial stream, **converts each polar reading into Cartesian XYZ coordinates** (simulating forward movement between slices), and renders a **live 3D point cloud** in Open3D.

CHALLENGES AND LESSONS

- Flat 2D slices couldn't form a volumetric map; added a simulated forward-translation step in the Python visualizer to stitch slices into a 3D cloud.
- Sensor and stepper motor were out of sync; implemented a firmware state machine to trigger VL53L1X reads immediately after each 11.25° step.
- Pushbutton bounce caused false scan toggles; implemented **debounce logic** in firmware to ensure clean start/stop commands