Individual Lab Report (ILR02)

Motor Lab

Name: Hongru Liu

Team: Team 2

Andrew ID: hongrul

Teammates: Hongru; Yechen; Hongfei; Dan; Bi

Submission Date: [2026-02-05]

Individual Progress

During this lab period, I focused on developing a standalone laptop GUI to satisfy the Motor Control Lab requirement. I implemented the GUI in Python using Tkinter for the user interface and pyserial for serial communication. The GUI includes serial port selection and connect/disconnect controls, a live telemetry panel to display three sensor readings and the three motor states (servo angle, DC motor position/velocity, and stepper position), and a command panel to send motor setpoints. The GUI also provides mode controls to support system state transitions (sensor control, GUI control, and stop/safe state).

To support integration with the single-program microcontroller firmware, I defined a simple serial protocol. The GUI expects telemetry messages from the microcontroller in the format:

`STATE,s1,s2,s3,servo_deg,dc_pos_deg,dc_vel_dps,stepper_pos_deg,mode`

and it sends newline-terminated command messages such as SER,<deg>, STP,<deg>, DC,POS,<deg>,VEL,<deg/s>, and MODE,<m>.

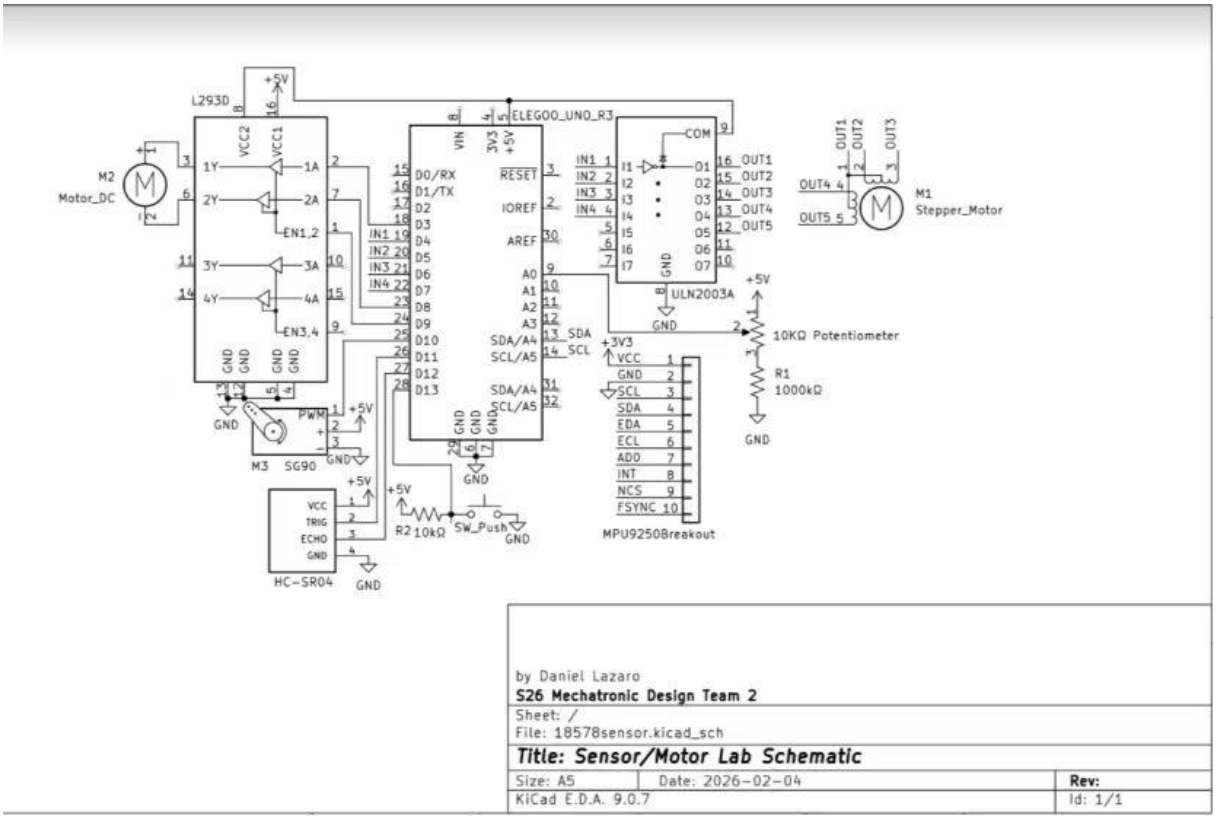

Figure 1. Sensor/Motor Lab schematic showing Arduino, motor drivers, motors, sensors, and switch connections.

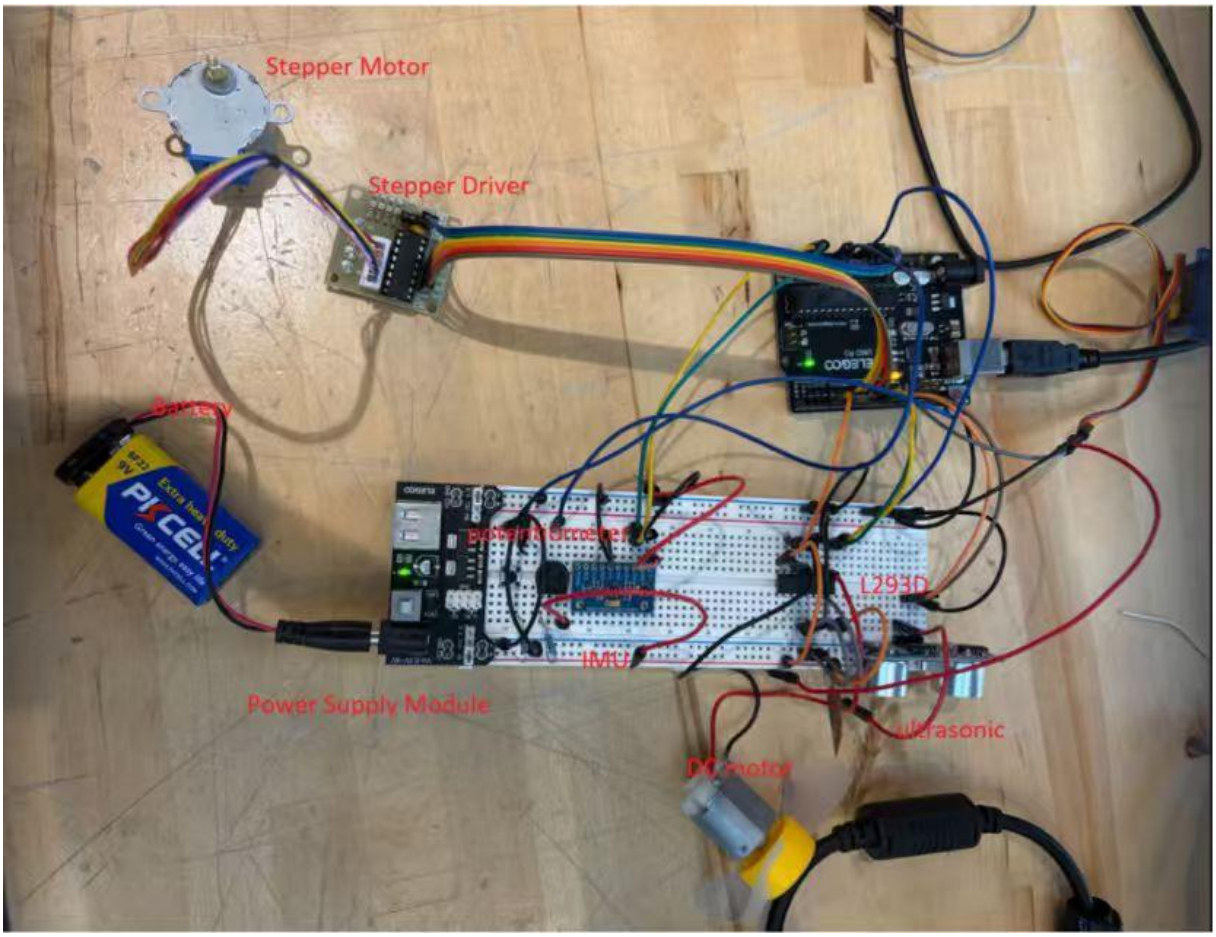

Figure 2. Bench-top prototype setup showing stepper motor + driver, DC motor + driver, sensors, and power/wiring used for the lab demonstration.

Challenges

The challenge is that the GUI is complete as a standalone application, but full end-to-end functionality depends on firmware integration: the microcontroller only run a single integrated program that continuously streams the expected STATE telemetry and parses the GUI command messages to update motor setpoints. Before integration is finalized, the GUI only demonstrated communication structure and interface design, but not full closed-loop control for all three motors simultaneously.

Teamwork

Team tasks were divided to progress. Hongfei focused on developing and testing the Arduino code for the DC motor and stepper motor subsystems. Dan focused on circuit wiring and

hardware debugging for the DC motor driver and encoder signals, and he also worked on integrating sensor logic with motor control. Yecheng implemented and tested the RC servo behavior, including verifying range of motion and response to command inputs. I focused on implementing the Python GUI, defining a consistent serial protocol, and providing a framework to monitor and command motor states during the demonstration.