Lab 2 - Introduction to sensors and actuators

Sensors and actuators form the I/O "muscles and senses" of any embedded system. Sensors translate real-world phenomena—light intensity, temperature, force, motion—into electrical signals that a microcontroller can sample, filter and interpret. Actuators perform the reverse conversion: they take the microcontroller's digital decisions and turn them into physical action by driving LEDs, buzzers, motors, solenoids or valves. When you connect a distance sensor to an Arduino Uno or mount an RGB color sensor on the Alvik robot, you're giving the code factual insight about its environment; when that code modulates a servo position or wheel speed, you're watching software directly shape hardware behavior. Well-designed embedded products close this sense–think–act loop at millisecond rates, enabling everything from precise motor control in drones to energy-efficient smart lighting. Selecting the right sensor–actuator pair, conditioning their signals correctly and timing their interaction reliably are therefore core skills for any embedded engineer.

1. Getting Started – Hands-On References

Activity	Quick Link / Where to look
Motion: precise forward & reverse driving	*There and Back Again* - https://courses.arduino.cc/explore-robotics-micropython/lessons/there-and-back-again/
Distance sensor	*Robot Dozer* - https://courses.arduino.cc/explore-robotics-micropython/lessons/robot-dozer/
IMU orientation basics	Need More Power - https://courses.arduino.cc/explore-robotics-micropython/lessons/need-more-power/
Color sensor: detect & react to colors	*Colorful Reactions* - https://courses.arduino.cc/explore-robotics-micropython/lessons/colorful-reactions/
Line-following sensors: IR array practice	*Walk the Line* - https://courses.arduino.cc/explore-robotics-micropython/lessons/walk-the-line/
Servo actuator demo code	`servoExample.py` (ExampleCodes folder)
Capacitive touch-pad demo code	`touchExample.py` (plus timing version `touchTimerExample.py`) (ExampleCodes folder)
Additional sample projects & libraries	Additional MicroPython example set → https://github.com/arduino/arduino-alvik-mpy/tree/main/examples

Use this table as your quick-reference hub

2. Motion: precise forward & reverse driving

Getting Started with Alvik tutorial:

- 2_1- There and Back Again Explore Robotics in MicroPyton.pdf

Or use the online material available at: https://courses.arduino.cc/explore-robotics-micropython/

Lessons→Motion and Turning → *There and Back Again* https://courses.arduino.cc/explore-robotics-micropython/lessons/there-and-back-again/
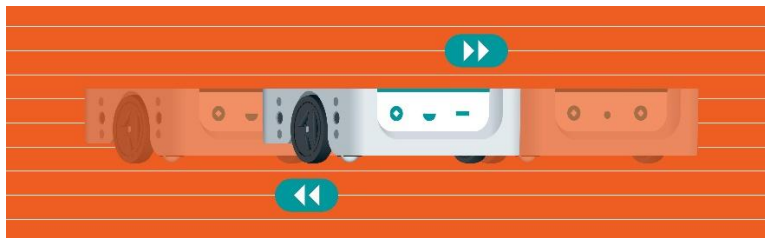

Figure 1: Alvik Motion - https://courses.arduino.cc/explore-robotics-micropython/lessons/there-and-back-again/

3. Distance sensor

Getting Started with Alvik tutorial:

- 3_1-Robot Dozer Explore Robotics in MicroPyton.pdf

Or use the online material available at: https://courses.arduino.cc/explore-robotics-micropython/

Lessons→Automation→ Robot Dozer https://courses.arduino.cc/explore-robotics-micropython/lessons/robot-dozer/

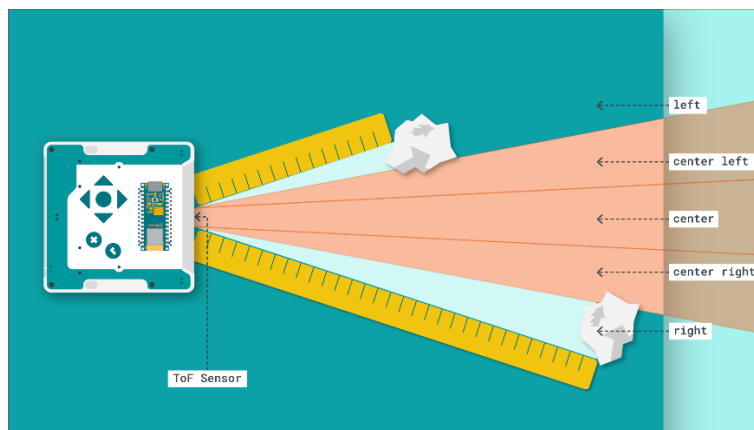*Figure 2: Alvik TOF Sensor -https://courses.arduino.cc/explore-robotics-micropython/lessons/robot-dozer/*

4. Additional examples:

Use any remaining lab time to **dig into the extra manuals or the sample scripts**. Each showcases a single sensor or actuator

For the Lab report:

1. Check-Your-Understanding Questions

   a. **Field-of-view math** The VL53L7CX ToF sensor reports an 8 × 8 ranging matrix. *How many individual distance readings per frame does that give, and why might a wide 90 ° FoV be useful for obstacle detection?* st.com

   b. **I²C address lookup** What default 7-bit I²C address does the VL53L7CX use on Alvik, and why is that important when you add a second external ToF sensor?

   c. **IMU full-scale choice** List two situations where you would switch the LSM6DSOX accelerometer from ±2 g to ±16 g. st.com

   d. **Active vs. passive sensing** Give one *active* sensor and one *passive* sensor on Alvik and explain the difference in one sentence.

   e. **Power-rail reasoning** Why must the micro-servo **not** be powered from the Nano ESP32 3V3 pin?

   f. **Alvik motion control** When you call:

      a. alvik.set_wheels_speed(+150, –150) # left_mm_s, right_mm_s

   What will the robot do, and how could you use the wheel-encoder counts to stop it after a 90 ° in-place right turn?

2. Post-Lab Reflection Tasks

   a. **Experience summary (≤150 words)** – Write a concise paragraph that touches on one "aha!" moment and one challenge you overcame.

   b. **Block diagram** – Draw a functional block diagram for one of the examples demonstrated in the manuals or the additional examples reviewed. Label each connection with its interface (I²C, UART, PWM, GPIO) and data direction.

   c. **Sensors summary (½ page)** – prepare a summary of the sensors used with the Alvik platform and their usefulness. Include part number and specifications for each of the sensors. You can find part numbers under the Alvik specifications page: https://docs.arduino.cc/hardware/alvik/#tech-specs