

Software Architecture Layout

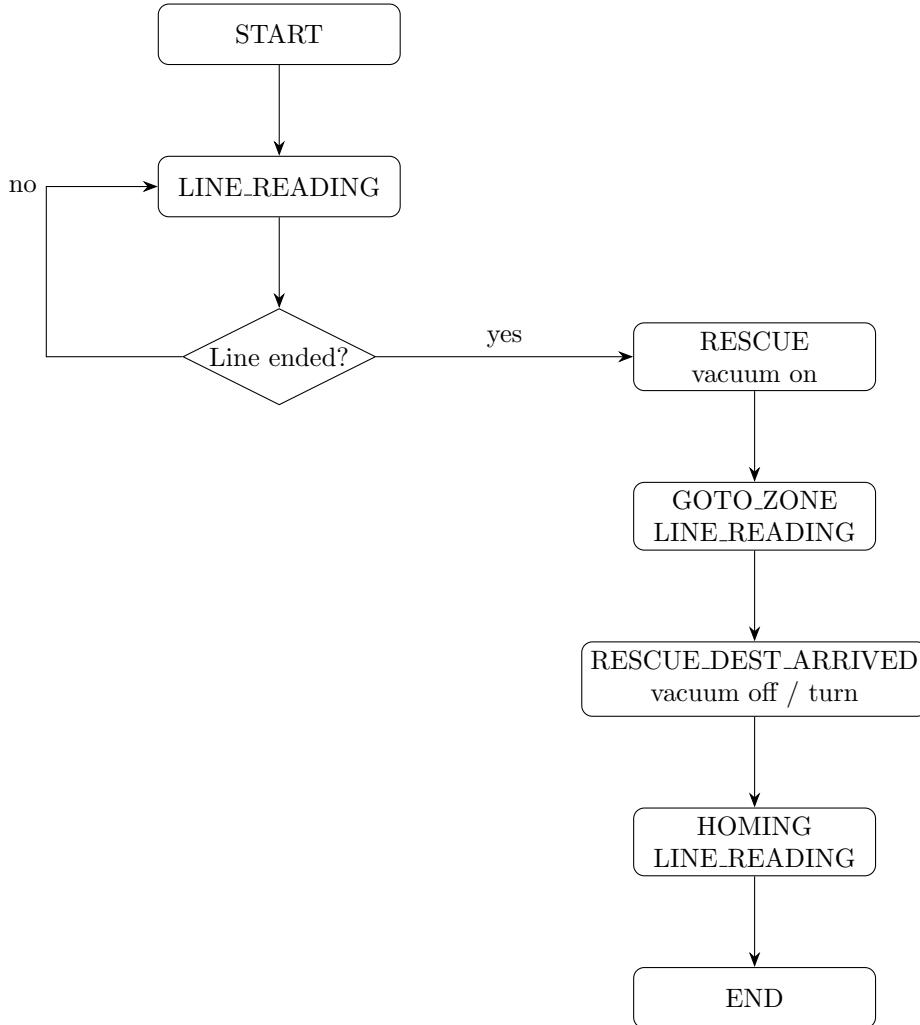
1 Software Plan

Using camera + IR/color sensor combination.

Infrastructure

- Use a state machine to track the current program stage.
- Start in LINE_READING, which fuses IR array signals and the camera frame to compute motion commands.
- If only one line is detected: calculate the line length L_k .
- Else: calculate the intersection first, then compute the line length L_k .
- Use L_k and the current speed to obtain the fuzzy mode and PID parameters.

2 State Machine Flowchart



3 Node-by-Node Behavior

- **START:** initialize sensors, camera, communication, and motor driver; set initial mode and variables.
- **LINE_READING:** run the closed-loop line follower by fusing IR error and camera line-length information to generate left/right motor commands.
- **Line ended?:** decision node; if the main line is no longer detected, transition to rescue sequence, otherwise continue LINE_READING.
- **RESCUE:** stop or slow as required, activate the vacuum mechanism, and secure the Lego figure.
- **GOTO_ZONE / LINE_READING:** navigate to the drop-off zone using the same line-reading controller (IR + camera + fuzzy/PID scheduling).
- **RESCUE_DEST_ARRIVED:** perform destination action (optional turn based on zone orientation), then deactivate vacuum to release the Lego figure.
- **HOMING / LINE_READING:** follow the return path home using line reading again, with mission-complete constraints if needed (e.g., reduced speed).
- **END:** stop motors, place actuators in safe state, and terminate or idle.

4 LINE_READING Loop (Camera + IR Fusion)

```
loop:
    ir = read_IR_array()
    err = line_position(ir) - center
    pid_u = PID(err, Kp, Ki, Kd)

    frame = camera_capture()
    lines = detect_line_vectors(frame)
    Lk = compute_primary_line_length_or_intersection(lines)

    X1 = speed_percent(current_speed)
    X2 = lookahead_percent(Lk)

    mode = fuzzy_or_table(X1, X2)
    baseSpeed, (Kp, Ki, Kd) = mode_parameters(mode)

    left = clamp(baseSpeed - pid_u)
    right = clamp(baseSpeed + pid_u)
    set_motors(left, right)
```

5 Notes

- X_1 is the current speed command normalized to 0–100.
- X_2 is the visible line length percentage computed from the camera ROI.
- The state machine transitions to RESCUE when the line ends, then to GOTO_ZONE, RESCUE_DEST_ARRIVED, and HOMING.