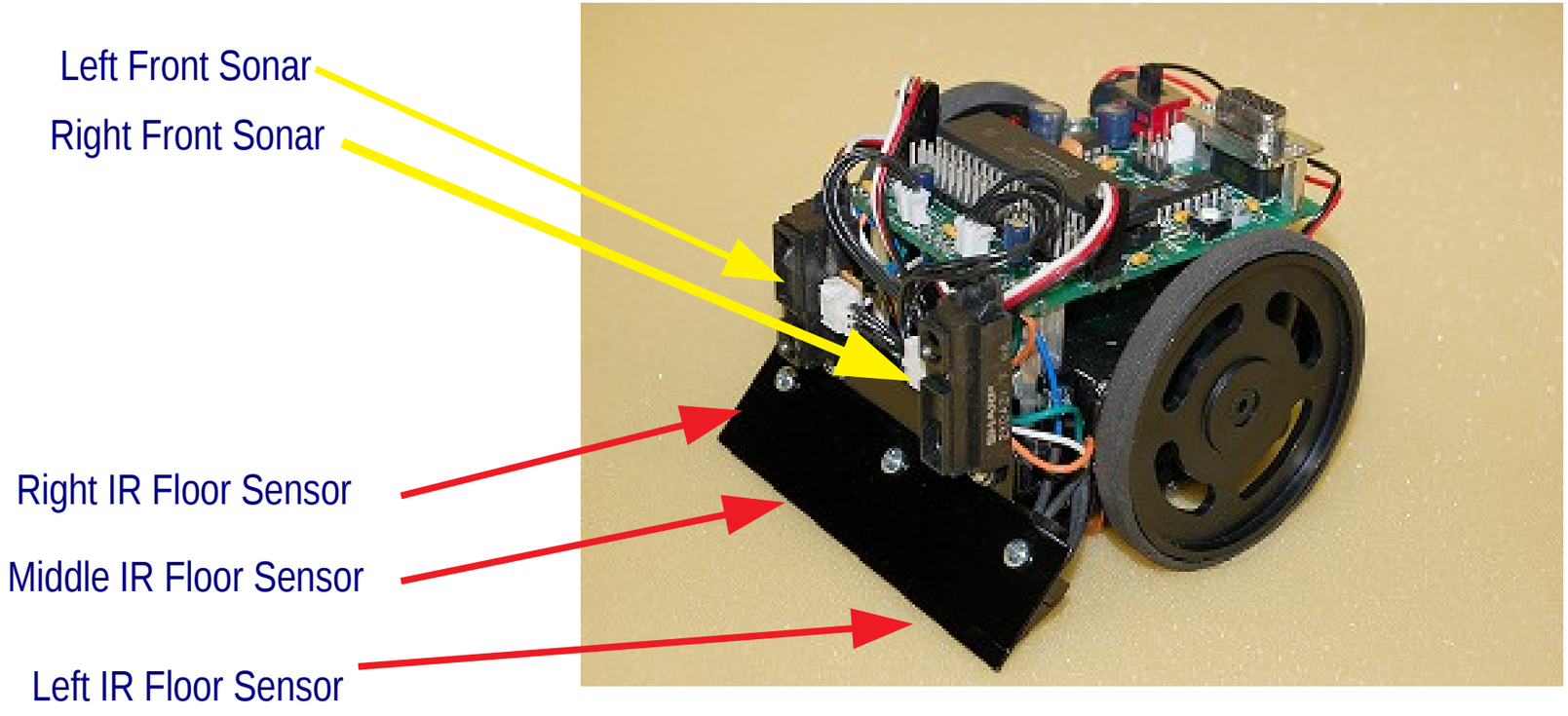
CS5000: F23: Theory of Computability

Supplementary Materials to Assignment 01: Problems 1 and 2
Robot Line and Blob Following FA

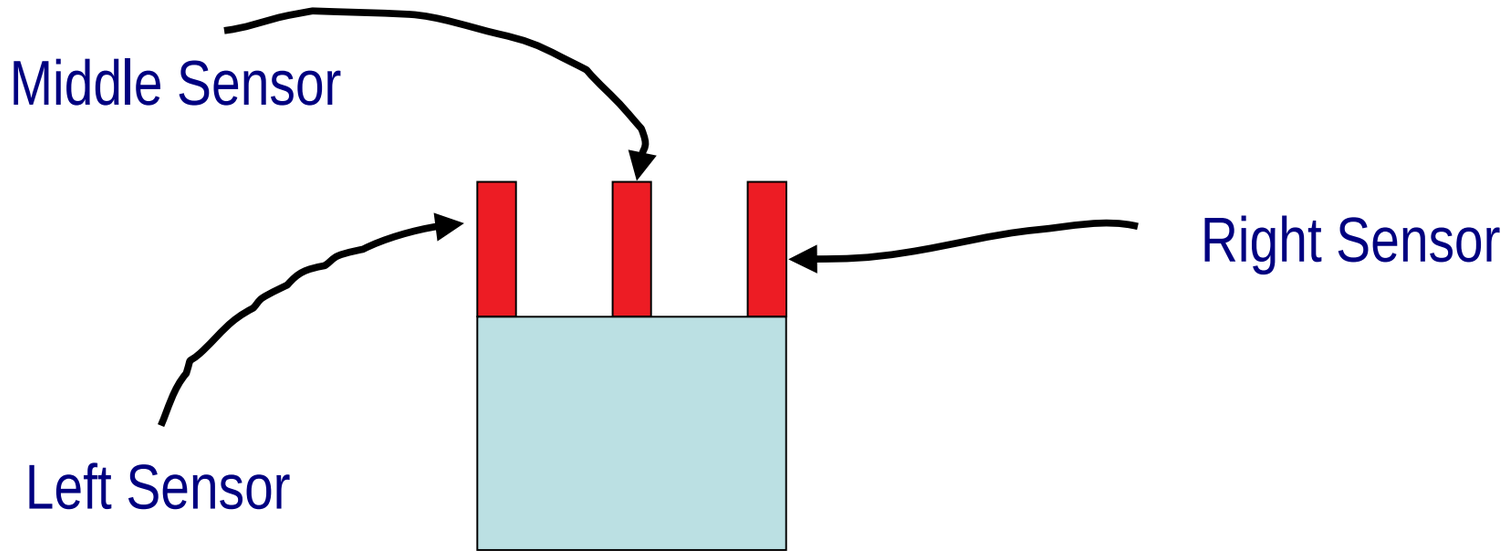
Vladimir Kulyukin
Department of Computer Science
Utah State University

Junun Mark III Robot' Sensors



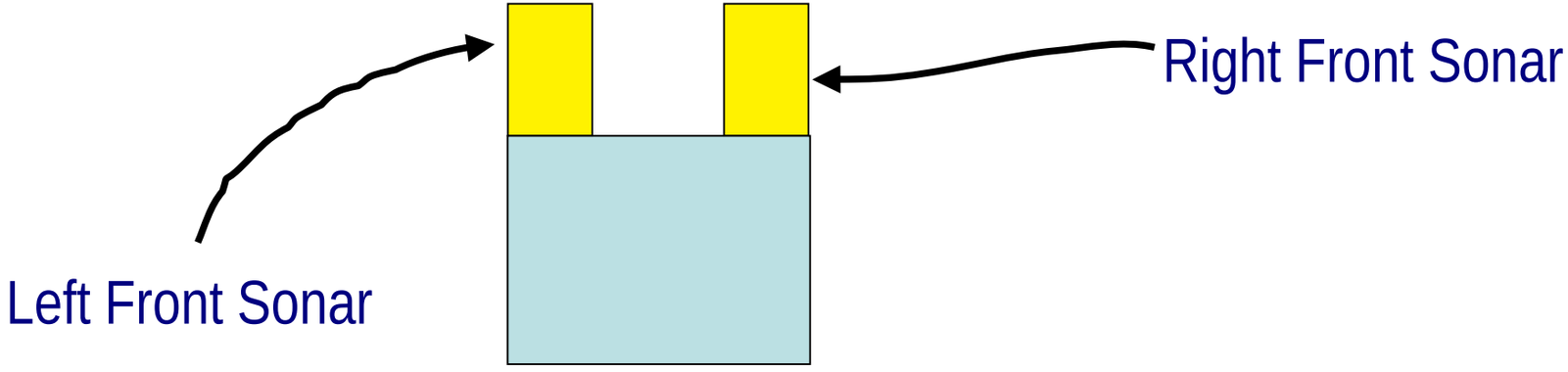
This robot has the 16f84a chip. The chip runs programs written in 16f84a Assembler.

Calibration of Junun Robot's IR Floor Sensors



A Junun robot has 3 IR floor sensors (left, middle, rgiht); each sensor can be calibrated to return two logical values: *black* and *white*

Calibration of Junun Robot's Front Sonars

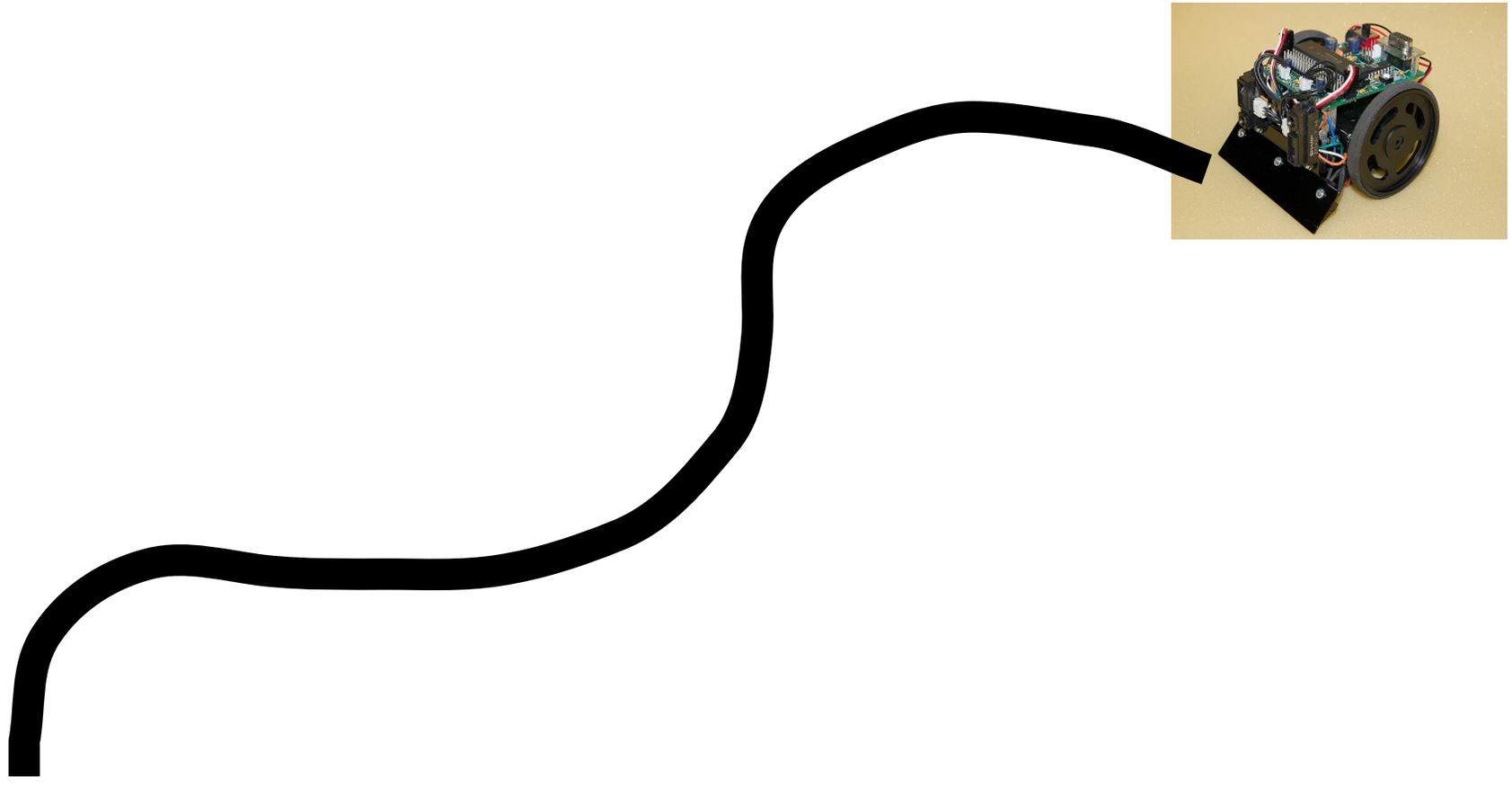


A Junun robot has 2 front sonars (left, right); each sonar can be calibrated to return two logical values: *obstacle* and *no_obstacle*

Problem 1: Robot Line Following Problem

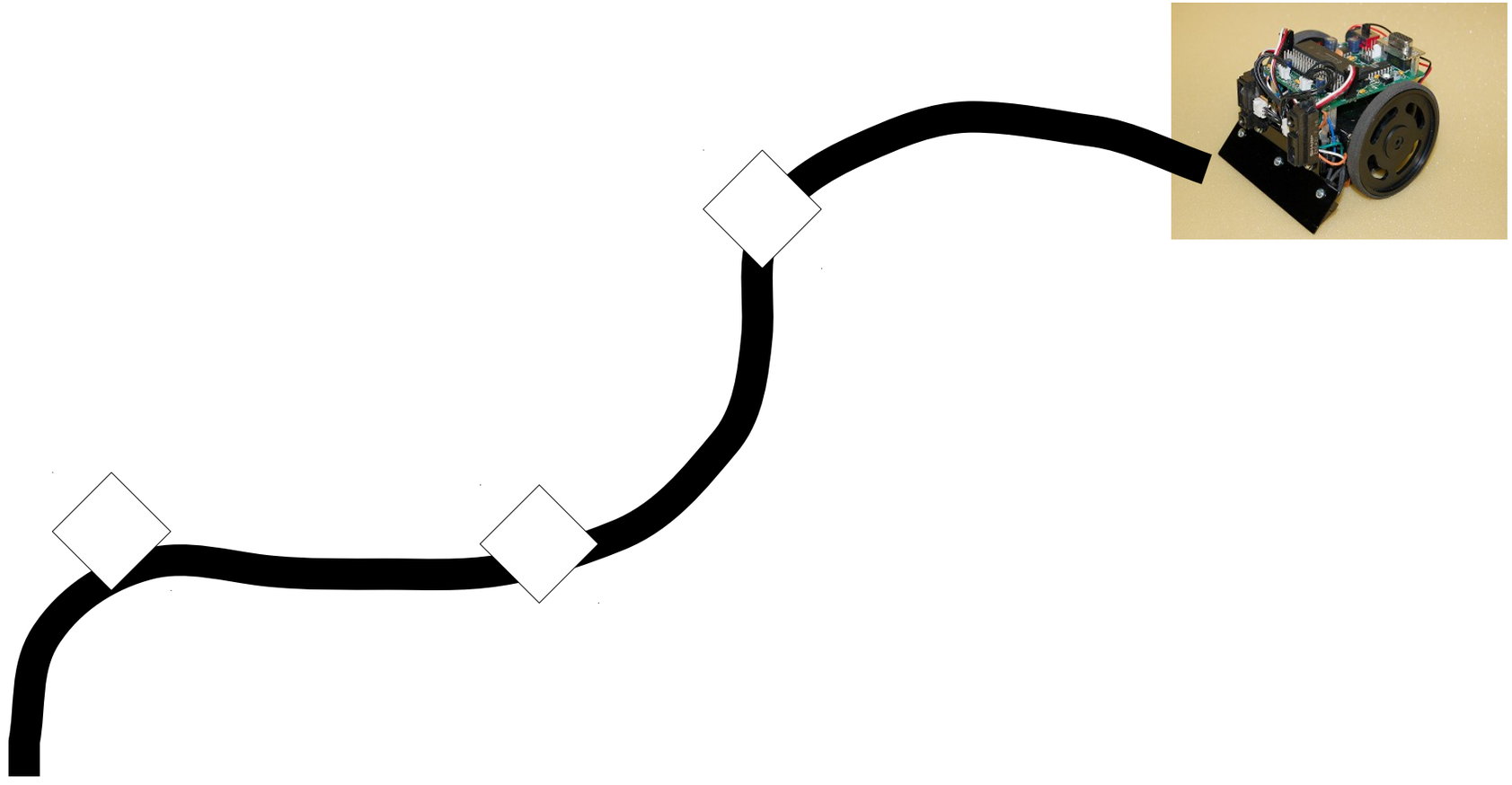
Design an FA to have a Mark III robot follow a line on the floor

Line Following Problem: Follow Line Start to End



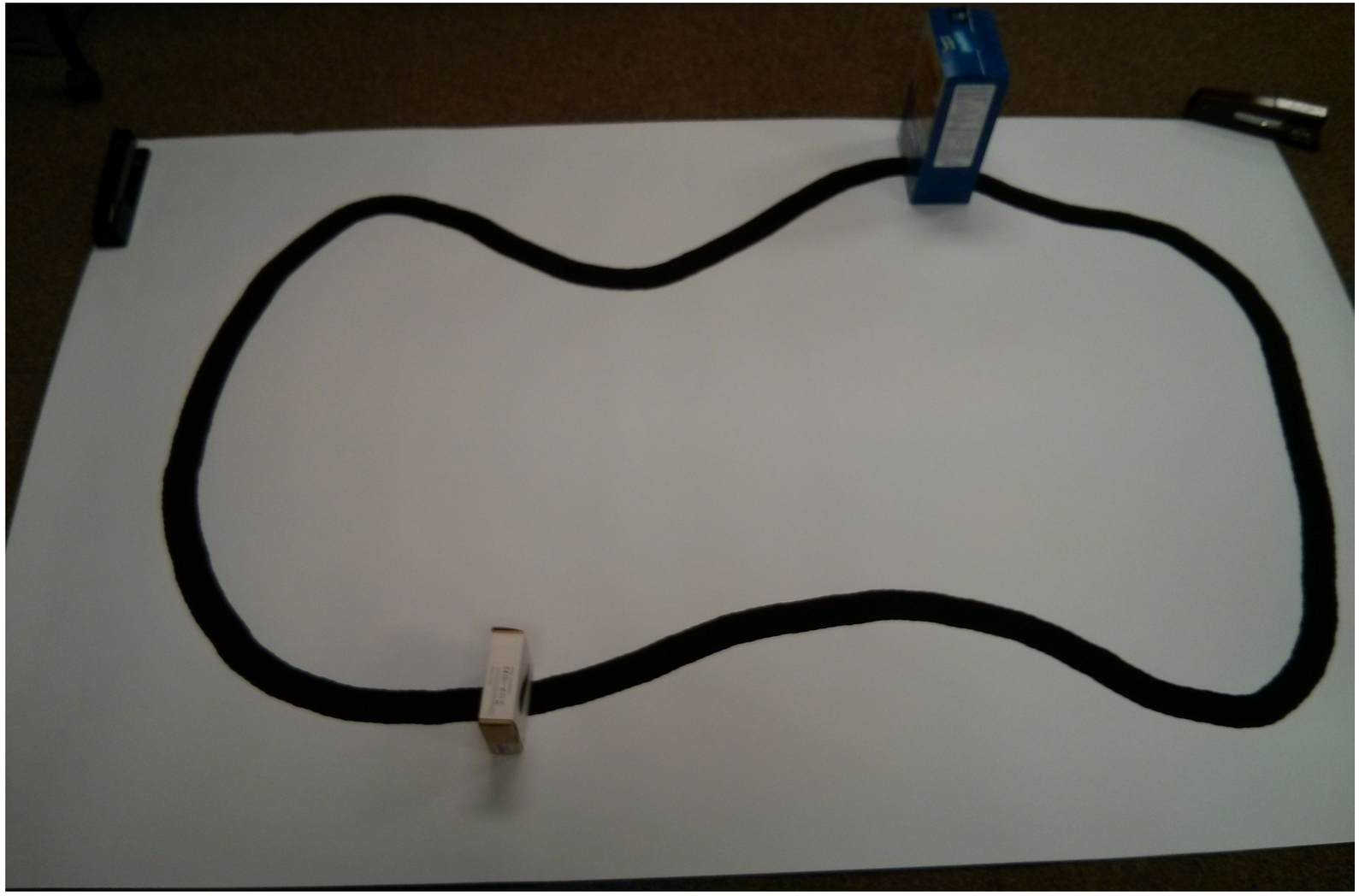
Follow line from start to end if there is an end; if there is no start or end (e.g., the line is the contour of a polygon) keep following it.

Line Following with Obstacles: Follow Line & Avoid Obstacles



A harder version of the line following problem – follow the line and avoid obstacles

A Sample Line Following Course for Junun Mark III



Hardware States of Line Following Robot

While following the line, the robot's hardware can be in one of the following states:

1) **read_sensors** – in this state the robot reads its 3 IR floor sensors; this is the start state of the robot;
2) **turn_left** – in this state the robot executes a small turn left (e.g., 10 degrees);
3) **turn_right** – in this state the robot executes a small turn right (e.g., 10 degrees);
4) **move_forward** – in this state the robot moves a small distance forward (e.g. 2cm).

Hardware Signals of Line Following Robot

While following the line, the robot's hardware issues the following signals from each state:

1) **read_sensors** – **black_on_left** (left IR detects black); **black_on_right** (right IR detects black); **black_on_middle** (middle IR detects right); white (no blackness is detected on any of the 3 IR sensors);
2) **turn_lef**t – **left_turn_complete** (hardware has completed a small left turn);
3) **turn_right** – **right_turn_complete** (hardware has completed a small right turn);
4) **move_forward** – **move_complete** (hardware has completed a small move forward);
5) For simplification, we can use **move_complete** instead of **left_turn_complete** and right_turn_complete

Mapping Hardware States and Signals to FA States and Signals

The robot's hardware states naturally map to the FA's states. In other words, the FA that controls the line following robot has the following set of states **Q**: **read_sensors**, **turn_left**, **turn_right**, **move_forward**; formally,

Q = {turn_left, turn_right, move_forward, read_sensors}

The robot's hardware signals naturally map to the alphabet symbols (i.e., the edge labels) between the FA states. In other words, the FA's alphabet (Σ) has the following complex symbols: black_on_left, black_on_right, black_on_middle, white; formally,

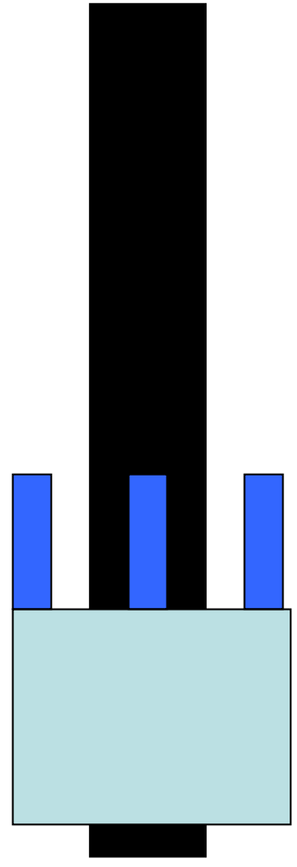
Σ = {black_on_left, black_on_right, black_on_middle, white, move_complete, left_turn_complete, right_turn_complete}

Of course, we can abstract away these details and use the standard FA notation (e.g,.

q_0 for **read_sensors**, **a** – for **black_on_left**, etc.)

Line Following: MOVE_FORWARD

```
if ( black_on_center )
    move_forward;
```



It is assumed that the robot starts with the middle sensor aligned with the line. So, if the middle sensor detects blackness, the robot moves forward.

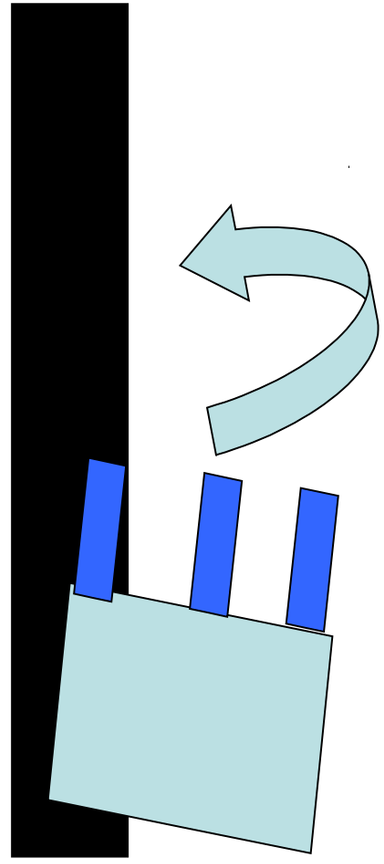
LEFT

CENTER

RIGHT

Line Following: TURN_LEFT

```
if ( black_on_left )
    turn_left;
```



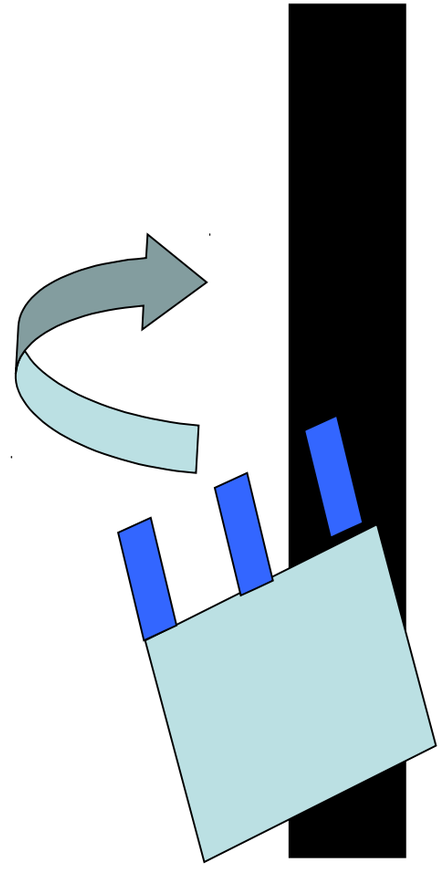
LEFT

CENTER

RIGHT

Line Following: TURN_RIGHT

```
if ( black_on_right )
    turn_right;
```



LEFT

CENTER

RIGHT

Problem 2: Robot Blob Following Problem

Design an FA to have a Junun robot follow a blob with its front sonars

Robot Blob Following Problem

The blob following problem is simpler than the line following problem. All the robot has to do is to follow the blob (i.e., obstacle) with its front sonars.

The FA for this problem will have the same states as the FA for the line following problem.

The only difference is that in the state **read_sensors**, the robot's hardware will issue the following signals: **obstacle_left** (left sonar detects obstacle); **obstacle_right** (right sonar detects obstacle); **obstacle_both** (both sonar detect obstacle); **obstacle_none** (neither sonar detects obstacle).

Robot Blob Following Problem

When following the blob, the robot starts in the **read_sensors** state.

If neither sonar detects obstacle, the robot stays motionless in the **read_sensors** state.

If both sonars detect obstacle, the robot moves forward.

If the right sonar detects obstacle, the robot moves right.

If the left sonar detects obstacle, the robot moves left.