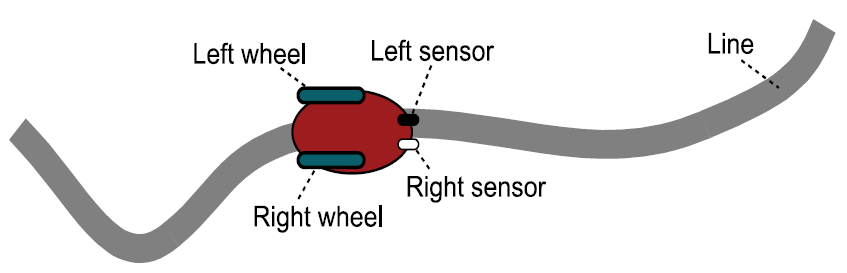
Finite State Machines - Exercise

# Objective

Design the controller of a line follower robot using the Finite State Machine concept.



# Definitions

From the point of view of your Controller:

* Inputs = 2 Sensors

|  |  |  |
| --- | --- | --- |
| Left Sensor value | Right Sensor value | Description |
| 1 | 1 | On line |
| 1 | 0 | Off to the right |
| 0 | 1 | Off to the left |
| 0 | 0 | Lost |

* Outputs = 2 Actuators (motors)

|  |  |  |
| --- | --- | --- |
| Left Wheel value | Right Wheel value | Description |
| 1 | 1 | Go straight |
| 1 | 0 | Turn right |
| 0 | 1 | Turn left |
| 0 | 0 | Stop |

# To do

1. Define a state diagram where all possible states and their transitions are defined, taking into account the possible inputs and the corresponding outputs.
2. Write the code of ***controller.h*** and ***controller.c*** files corresponding to the diagram.
3. Add in the following ***main.c*** file all the code that is necessary to merge your state machine with the existing ones:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*   Libraries   \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <unistd.h>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*   State Machines   \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include "sensors.h"

#include "motors.h"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*   Main Program   \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main()

{

    printf("Main: Init \n");

    // Initialize all state machines

    InitializeSensors();

    InitializeMotors();

    // Infinite loop

    while(1)

    {

        // Update all state machines

        UpdateSensors();

        UpdateMotors();

        /\* This delay here is for learning purposes only \*/

        /\* In a real condition, a 1ms interrupt must be implemented

        and used to decrease a counter variable \*/

        usleep(1000);   // 1 ms

    }

    printf("Main: End \n");

    return 0;

}

1. Test the implementation by simulating the entire code. For the inputs, add a function that randomly returns values from the sensors. And for the output just print the state of the motors in the terminal.