

## **0.1 END OF LINE DETECTION**

### **0.1.1 Tasks:**

1. Stop the robot when it reaches the end of line.
2. Detecting the end of line can be done by measuring the time that robot spend over the black and white area. E.g. if the robot is driving along the line - the time spent over black and time spent over white area will be quite the same. When line ends the robot will not detect the black area soon and the time spent over white area will increase significantly - and that is the trigger for detecting the end of line.
3. Advanced: Make a function to align (move) the robot back to the line.

### **0.1.2 Questions:**

1. How can we store a data to the controller's memory?
2. How can we measure time in programming loops?
3. What is the purpose of the prog. instr. `exit(0);` ?

**Program 1:** Edn of Line Detection.

```
1  #include "RobotMovingFunctions.h"
2  const int LIGHT_SENSOR_PIN = A0;
3  const int SURFACE_BRIGHTNESS_REFERENCE = 400;
4  int time_on_black = 0;
5  int time_on_white = 0;
6
7  void setup()
8  {
9      setIOPins();
10     pinMode(LIGHT_SENSOR_PIN , INPUT);
11 }
12 void loop()
13 {
14     int light_sensor_value = analogRead(LIGHT_SENSOR_PIN );
15     if ( light_sensor_value < SURFACE_BRIGHTNESS_REFERENCE )
16     {
17         // BLACK area
18         moveLeft();
19         time_on_white = 0; // reset time on white
20         time_on_black++; // meas. time on black
21         delay(100);
22     }
23     else
24     {
25         // WHITE area
26         moveRight();
27         // Do similar meas.
28         // of time on white
29         delay(100) ;
30         // If time is signif. longer:
31         //      robotStop();exit(0);
32     }
33 }
```

**0.1.3 Summary:****0.1.3.1 <++>****0.1.4 Issues:****0.1.4.1 <++>**