EE 321 – Microprocessors
Term Project Final Report

MicroStrategy

Author	*Uygar Kaya* *Ahmet Batuhan Akkaya*
Instructor	*Prof. H. Fatih Uğurdağ*
Submission Date	*08.06.2022*

1 Introduction

In this course term project, we programmed a Zumo to scan the surroundings of the black arena with white borders and we report the number of the objects by blinking the led. By staying inside the arena throughout the process, the Zumo robot blinks the number of objects in the arena with the help of the LED 13 pin, by knocking down the surrounding objects if necessary or not.

More than one scenario was created for this project and these scenarios were tested at the necessary stages.

1.1 Hardware Used

1.1.1 Zumo Robot

The Zumo robot is a low-profile tracked robot platform intended for use with an Arduino (or compatible device) as its main controller.

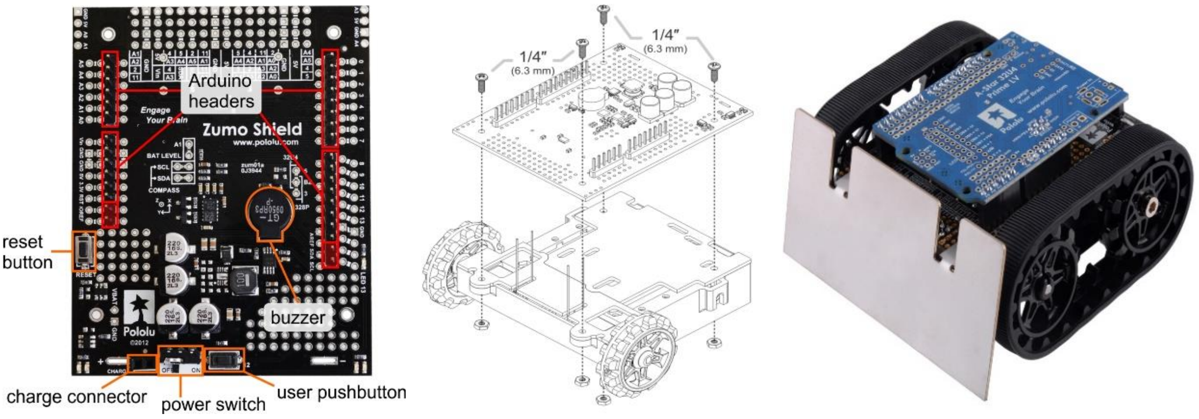

Figure 1: Schematic Representation of Zumo Robot with Arduino Connection [1]

1.1.2 Arduino

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write, and upload computer code to the physical board.

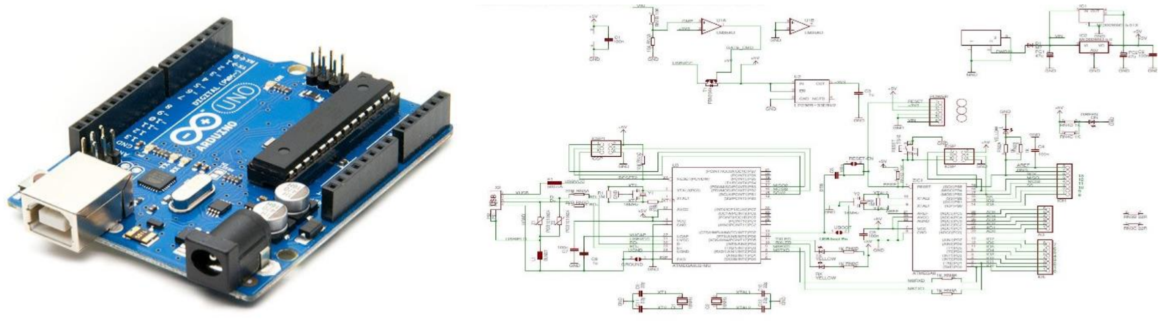

Figure 2: Arduino UNO and its Schematic [2]

1.1.3 MZ80 Infrared Sensor

MZ80 is preferred sensor for object detection. At back side sensor's range can be decreased or increased (10-80 cm for White Surfaces) with turning trimpot head screw.

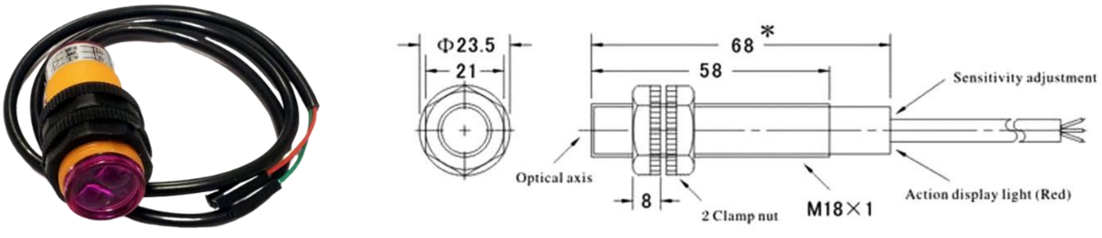

Figure 3: MZ80 Infrared Sensor and its Schematic

2 Background

This section explains how and for what tasks the Zumo robot is used in different projects.

We can do projects in many different fields using the Zumo robot. Border Detect, Line Follower, Proximity Sensor Test, Maze Solver, Object Tracking, RC Control, Sumo Collision Detect, and Mini Sumo can be given as an example of these projects.

If we are to explain a few of these examples, the Object Tracking project first searches for the object and when it finds the object, it locks onto the object and tracks the object in real-time.

In the Mini Sumo project, the calibration is done first and then the two Zumo robots detect and fight each other, and the robot that falls from the first arena loses the round.

3 Methodology

In this term project, we programmed a Zumo Robot that scans the arena and counts objects with using Arduino.

During this term project, using only 3 different Arduino programming language libraries were used these are;

1. *QTRSensors.h*
2. *ZumoReflectanceSensorArray.h*
3. *ZumoMotors.h*

QTRSensors library is used to interfacing with QTR Reflectance Sensors. I used to ZumoReflectanceSensorArray library to detect the white borders. Finally, I used the ZumoMotors library for PWM-based speed (and direction) control of the two motors on the Zumo.

4 State Machine of the System

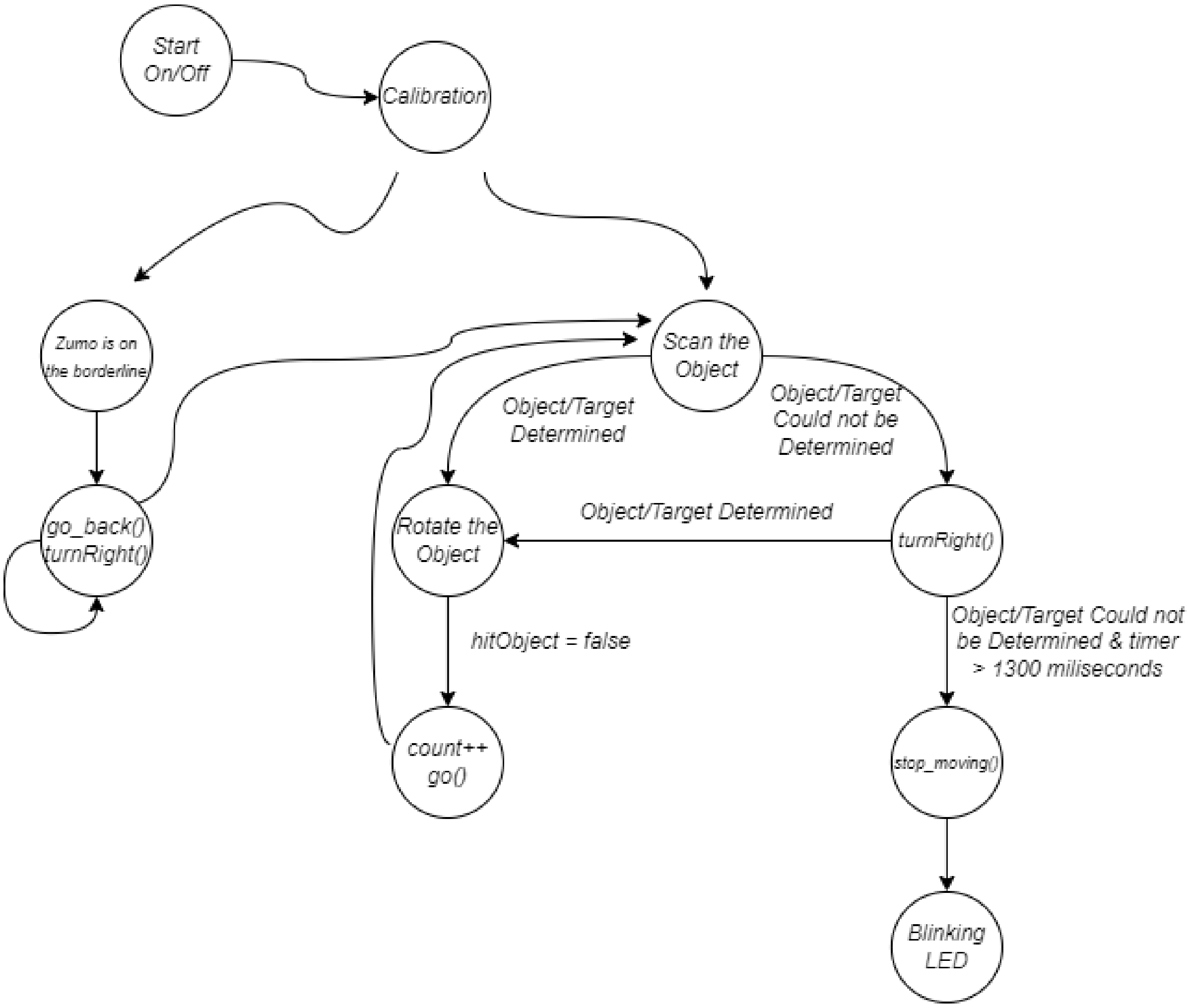

Figure 3: State Diagram of the System

5 Implementation Details

In this assignment, firstly we import the necessary libraries which are QTRSensors, ZumoReflectanceSensorArray, and ZumoMotors. After importing these libraries, we define the sensor pins and led pins. First of all, we make the setup method, in this method, we make the calibration for 10 seconds in order to separate the black and white points in the arena. After making the calibration, we first check whether the Zumo robot is at the border of the arena from the data we received from the sensor under the Zumo robot, if it is at the border, I take the Zumo robot back with the go_back() method and we determine the object around itself with the turnRight() method. If Zumo is not at the border, it scans the object and when it detects the object, it drops the object with the go() method and increases the count variable one by one, and then scans the object again, if it cannot find the object, it turns right around itself with the turnRight() method and scans the object.

In the last part, if the Zumo robot does not scan the object and rotates to the right for 1300 milliseconds, the Zumo Robot stops with the stop_moving() method and blinked the LED 13 as much as the count number.

6 Design of Experiments

In this project, we design a set of experiments. In each experiment, we test a different scenario. In this scenario, there are different number of objects, we set different angels from the center and different distances and corner cases.

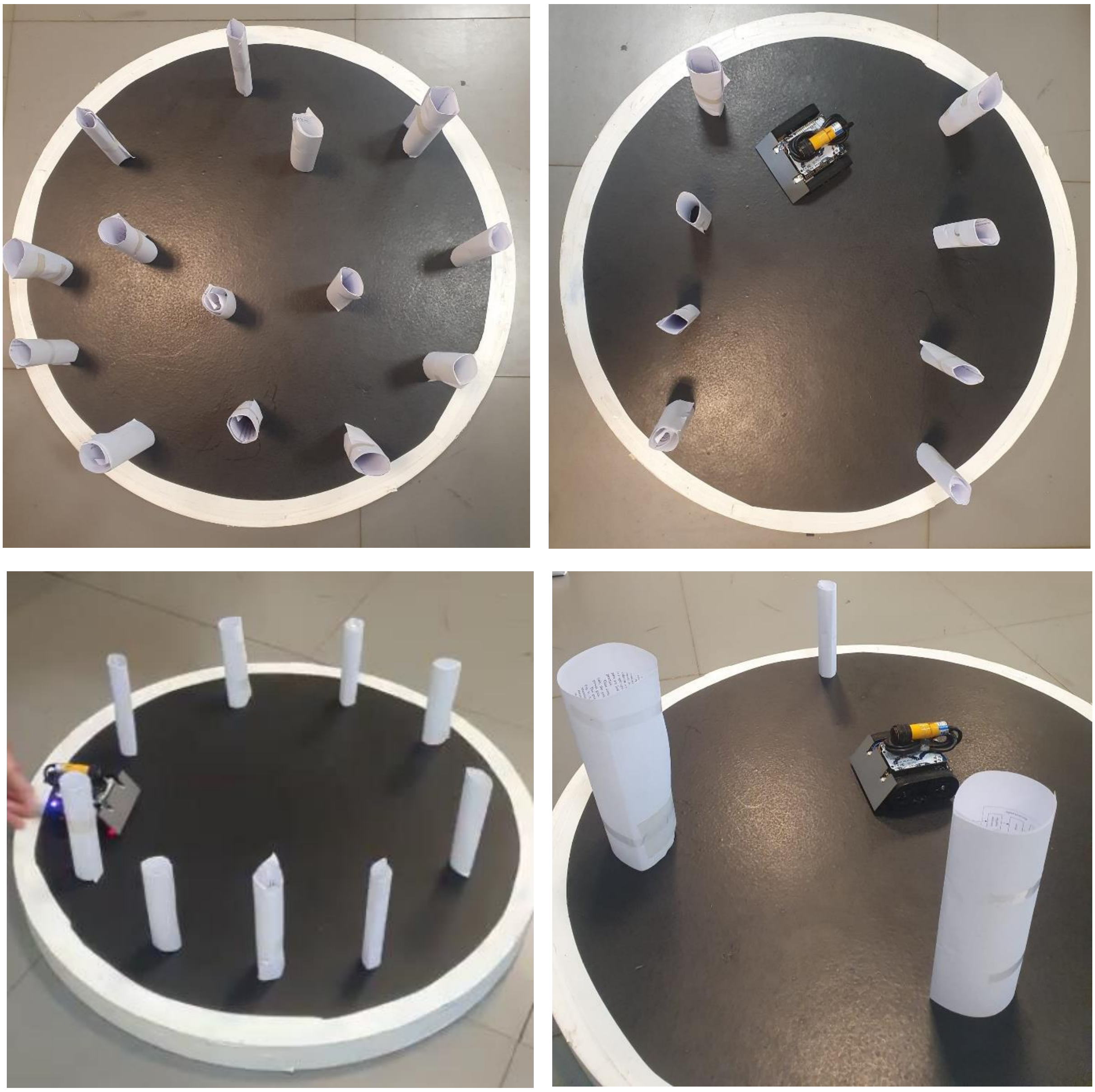

Figure 4: Different Test Scenarios of the Project

As you can see in the figure, we test our seniors with the different test parameters which are brightness of the area, number of the objects, angles of the objects, distance of the objects, size of the area, and size of the objects.

Furthermore, we test the speed settings of the Zumo robot and fix them at an optimum level. While making the speed settings of the Zumo robot, we tested the forward speed, reverse speed, and right rotation speed at an optimum level and set a fixed value that we determined for each value.

7 User Manual

1. ***Preparation***

   In this field, firstly, we need to make certain preparations to operate our Zumo robot. To power the Zumo robot and ensure that the sensor's range works at the maximum level, unused batteries must be taken and integrated into the Zumo robot.

2. ***Power – On / Power – Off***

   To run the Zumo robot, we first need to use the Power - On/Power - Off button. We should use the Power-On button to start the Zumo robot, and the Power-Off button after completing all our processes.

3. ***Making Calibration***

   We first perform a 10-second calibration for the Zumo robot to detect the borders.

4. ***Setting the Location***

   After calibrating, we leave it to an optimum location in the Zumo robot arena, and the Zumo robot knocks down the objects in the arena and blinked the LED as much as the number of objects it knocked over.

8 References

[1] *Pololu Zumo Shield for Arduino User's Guide. Pololu Robotics &amp; Electronics. (n.d.). Retrieved May 21, 2022, from https://www.pololu.com/docs/0J57/all*

[2] *Team, T. A. (n.d.). What is Arduino? Arduino. Retrieved May 21, 2022, from https://www.arduino.cc/en/Guide/Introduction*