EE 321 – Microprocessors
Term Project Progress Report

MicroStrategy

Author	*Uygar Kaya* *Ahmet Batuhan Akkaya*
Instructor	*Prof. H. Fatih Uğurdağ*
Submission Date	*29.05.2022*

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.

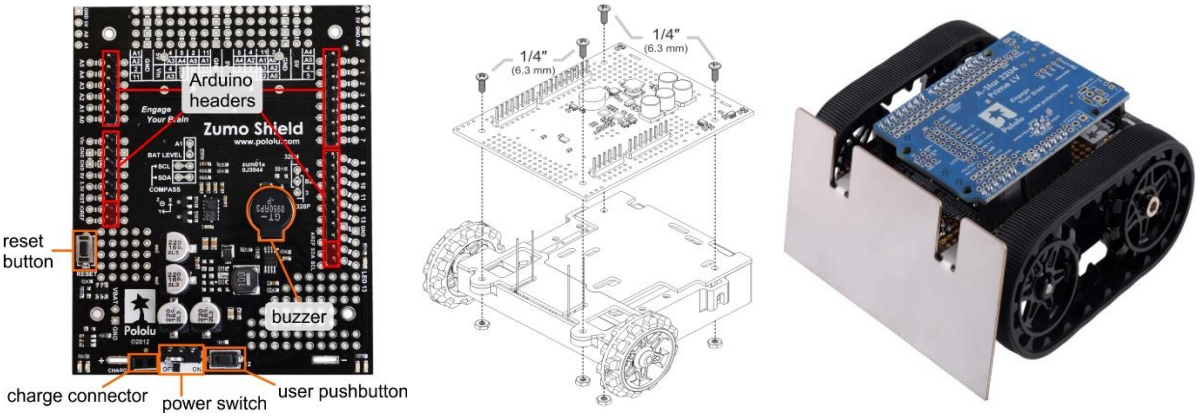


Figure 1: Schematic Representation of Zumo Robot with Arduino Connection [1]

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.

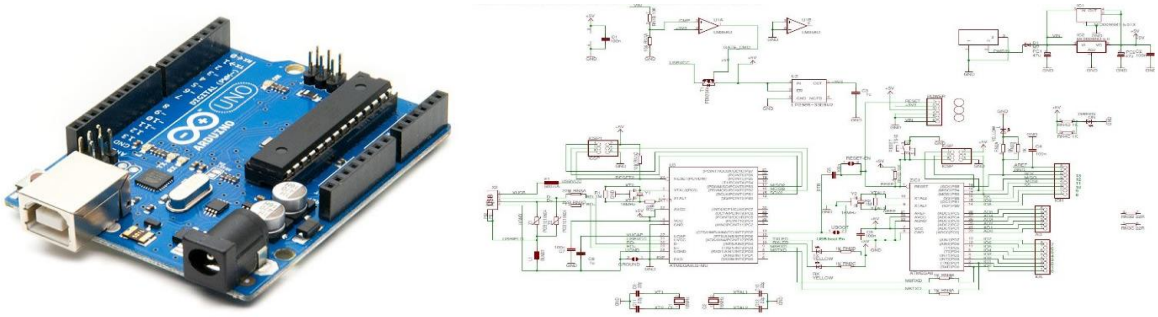


Figure 2: Arduino UNO and its Schematic [2]

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.

2 Methodology

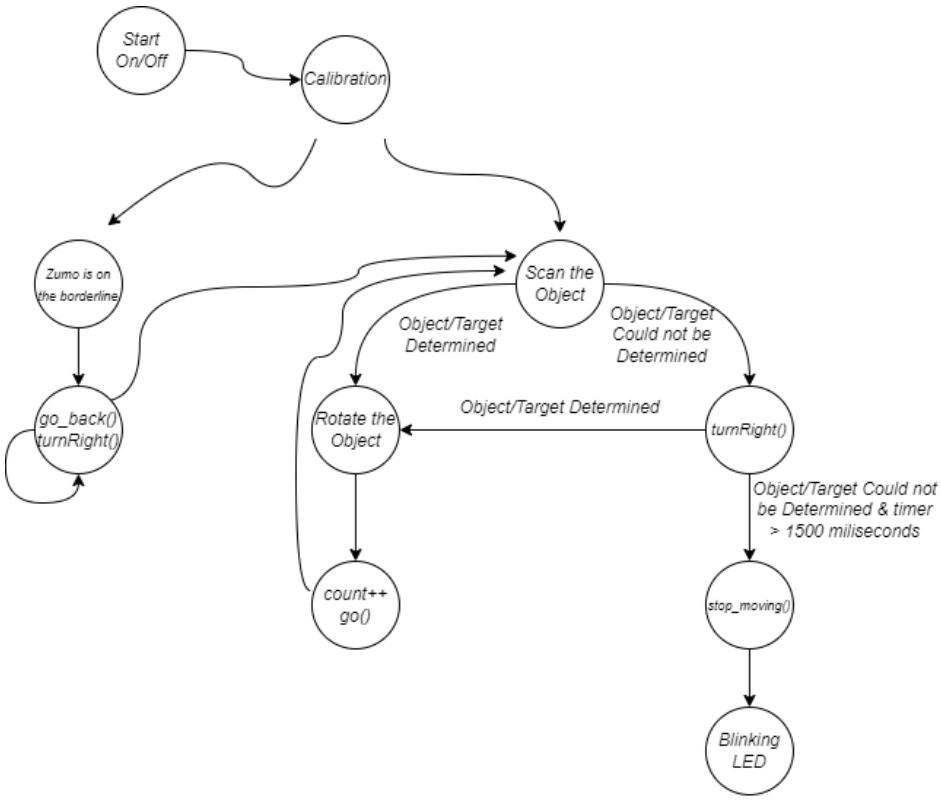
In this term project, we programmed a Zumo Robot that scans the arena and counts objects with using Arduino.

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.

3 State Machine of the System



4 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 1500 milliseconds, the Zumo Robot stops with the stop_moving() method and blinked the LED 13 as much as the count number.

5 Source Code

```
#include <QTRSensors.h>
#include <ZumoReflectanceSensorArray.h>
#include <ZumoMotors.h>

#define LED_PIN 13
#define MZ80_PIN 6
#define NUM_SENSORS 6

//Pushbutton button(ZUMO_BUTTON);
ZumoReflectanceSensorArray reflectanceSensors;
ZumoMotors motors;

// Define an array for holding sensor values.
unsigned int sensorValues[NUM_SENSORS];
unsigned int positionVal = 0;
unsigned int count = 0;
unsigned int timer = 0;
unsigned int hitObject = false;

void setup() {
  pinMode(LED_PIN, OUTPUT);
  digitalWrite(LED_PIN, HIGH);

  // --------------------------- Start Of The Calibration --------------------
  reflectanceSensors.init();
  unsigned long startTime = millis();
  while (millis() - startTime < 5000)  // make the calibration take 10 seconds
  {
    reflectanceSensors.calibrate();
  }
  // --------------------------- End Of The Calibration ----------------------
}

unsigned int mostLeftSensor() {
  if (sensorValues[0] < 600)
    return 1;
  else
    return 0;
}

unsigned int leftSensor() {
  if (sensorValues[1] < 600)
    return 1;
```

```
  else
    return 0;
}

unsigned int midLeftSensor() {
  if (sensorValues[2] < 600)
    return 1;
  else
    return 0;
}

unsigned int midRightSensor() {
  if (sensorValues[3] < 600)
    return 1;
  else
    return 0;
}

unsigned int rightSensor() {
  if (sensorValues[4] < 600)
    return 1;
  else
    return 0;
}

unsigned int mostRightSensor() {
  if (sensorValues[5] < 600)
    return 1;
  else
    return 0;
}

void turnRight() {
  motors.setSpeeds(130, -130);
}

void go() {
  motors.setSpeeds(880, 880);
}

void go_back() {
  motors.setSpeeds(-400, -400);
}

void stop_moving(){
  motors.setSpeeds(0, 0);
}
```

```
void loop() {
  positionVal = reflectanceSensors.readLine(sensorValues);
  if(mostLeftSensor() == 1) {
    go_back();
    delay(350);
    turnRight();
    delay(100);
  }
  else {
    if(digitalRead(MZ80_PIN) == 1) {
      timer = timer + 1;
      turnRight();
      if((timer >= 1500) && digitalRead(MZ80_PIN) == 1) {
        stop_moving();
        for(int index=0; index<count; index++) {
          digitalWrite(LED_PIN, HIGH);
          delay(1000);
          digitalWrite(LED_PIN, LOW);
          delay(1000);
        }
        timer = 0;
      }
      hitObject = false;
    }
    else {
      if(hitObject == false) {
        go();
        count = count + 1;
        hitObject = true;
        timer = 0;
      }
    }
  }
}
```

6 References

[1] *Pololu Zumo Shield for Arduino User's Guide. Pololu Robotics & 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*