TXL18S-1 B

Rukhsora Nazarova

Mikhail Stefantsev

Mariia Saveleva

# Robot projects

Table of Contents

[Robot projects 1](#_Toc529374296)

[1. Abstract 2](#_Toc529374297)

[1.1. /\* General summary of the whole project \*/ 2](#_Toc529374298)

[2. Introduction 2](#_Toc529374299)

[2.1. /\* The goal and scope of the project \*/ 2](#_Toc529374300)

[3. Methods and materials 2](#_Toc529374301)

[3.1. /\* What methods and materials are we using. Zumo, sensors, language, creator, libs \*/ 2](#_Toc529374302)

[4. Theoretical background 2](#_Toc529374303)

[4.1. /\* Some theory about how sensor, robot works, how some algorithms work \*/ 2](#_Toc529374304)

[5. Implementation 2](#_Toc529374305)

[5.1. Batteries 2](#_Toc529374306)

[5.2. Sensors 4](#_Toc529374307)

[5.2.1. Ultrasonic sensor 4](#_Toc529374308)

[5.2.2. Reflectance array 4](#_Toc529374309)

[5.2.3. Accelerometer 4](#_Toc529374310)

[6. Results 4](#_Toc529374311)

[6.1. /\* won or not, how was the challenges \*/ 4](#_Toc529374312)

[7. Discussion 4](#_Toc529374313)

[7.1. /\* What was challenging? Did we have problems? How did we solve them? Why did robot fail, for example? \*/ 4](#_Toc529374314)

[8. Conclusion 4](#_Toc529374315)

[8.1. 4](#_Toc529374316)

[9. References 4](#_Toc529374317)

## Abstract

* 1. /\* General summary of the whole project \*/

## Introduction

* 1. /\* The goal and scope of the project \*/

The goal of this project is to design and build a Zumo robot software that will allow it to pass all the challenges in the final assignment.

## Methods and materials

* 1. /\* What methods and materials are we using. Zumo, sensors, language, creator, libs \*/

## Theoretical background

* 1. /\* Some theory about how sensor, robot works, how some algorithms work \*/

The robot consists of three main part: Arduino, Zumo Shield and Zumo chassis.

## Implementation

* 1. Batteries

First challenge we faced was related to the batteries in the robot. Since Zumo uses conventional nickel metal hydride batteries that should never be discharged below the certain threshold in order to stay functioning we had to implement the function which constantly checks the voltage of the batteries and notifies the user in case of need to charge the batteries. When battery voltage gets too low, the robot starts blinking the onboard LED in full power that can’t be discarded in any way. It is also planned to implement the feature that locks the motors in case of low battery charge in order to prevent them from further discharging and subsequently damaging them. This required us to calculate the real voltage from the readings of the battery ADC.

The ADC connection diagram:



As seen from the diagram, the ADC input is connected to the output of the divider to lower the voltage as it may exceed the maximum allowed voltage of the ADC itself. That is why it is required to not only convert ADC output to volts, but also get the source voltage.

To convert the ADC output to the volts, the following conversion coefficient was used: , where is the reference voltage of the ADC (*5V as specified in the documentation)* and is number of bits used by ADC (*in our case, =12*).

To get the source voltage the voltage conversion coefficient is needed. It is calculated like:

where is the equivalent resistance of and .

So, the source voltage equals to:

where is the output level of ADC (*an integer number between and* ).

* 1. Sensors

Sensors can allow for a reliable robot operation in changing conditions such as changes in battery charge level, nonlinear motor output curve and other changing external conditions. Usage of sensors allows the robot to correct its actions in case of any changes due to which pre-programmed algorithm may not work as desired. Zumo shield has a few onboard sensors which can be used in a user-written program with the help of the Zumo library as described below.

* + 1. Ultrasonic sensor

Zumo robot can use an ultrasonic sensor to position itself relative to its surroundings. In this project the sensor will be used in the final task to detect maze walls. Zumo library provides the command to read ultrasonic sensor measurement in centimeters.

Zumo library includes a function called Ultra\_Start() which starts task handling sending ultrasound signal and interrupt receiving it and calculating the distance knowing the time to took the signal to reflect from the object in front of the robot and hit the sensor.

* + 1. Reflectance array

Zumo shield has a reflectance array consisting of 6 sensors located underneath the shield board behind the bulldozer blade on the front. These sensors are used in detecting a line for the robot to follow. Zumo library provides commands for reading both digital data from sensors using the threshold value and raw sensor readings.

On a top design level Zumo library uses a set of timers synchronized with SR switches. On function reflectance\_start() execution a separate task scanning sensor readings in the background is created and run every millisecond. The result of scanning can be written in the structure of type sensors\_.

* + 1. Accelerometer

The robot has a built-in accelerometer for detecting hits if the ultrasonic sensor did not detect the obstacle in advance which can be used in the sumo battle for detecting hits from sides and back where ultrasonic sensor cannot detect the approaching opponent robot. Zumo library provides function which allows to read the acceleration in , and direction.

## Results

* 1. /\* won or not, how was the challenges \*/

## Discussion

* 1. /\* What was challenging? Did we have problems? How did we solve them? Why did robot fail, for example? \*/

## Conclusion



## References

<https://www.pololu.com/product/2508>

30/11/18

Installed Bitbucked, Trello, got

01/11/18

Today is the second day with our robots. We are applying several commands to it. Such as giving values to motors (0 or 1). Make our robot to move forward and backward.